Personal SOFTWARE

ISSN. 0263-0605
SUMMER 1982 £1.95

A COMPREHENSIVE library of tested and documented software for the BBC Microcomputer

WELCOME
BBC microcomputer

Put your BBC micro to good use.

- Technical
- Games
- Financial
- Household
- Educational

From the publishers of computing today

BUG-BYTE SOFTWARE

CHESS

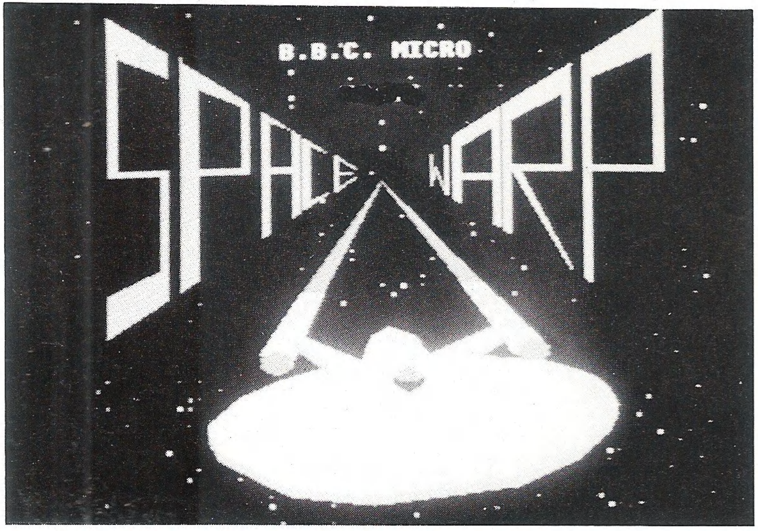
A fast version with many levels of play, using high resolution display. There are many options, including setting up, saving games on tape, changing level of play, castling and "en passen". Plays a very strong game. Price £11.50 inclusive. This program is suitable for the Model A(32K RAM) and the Model B.

GOLF

A very true to life simulation of an eighteen hole course. Complete with fairways, bunkers, greens and streams. Altogether a very compulsive game. Price £7.00 inclusive. Suitable for the Model A(32K RAM) and the Model B.

MULTIFILE

A multi-purpose filing system, with user definable set up. It caters for all file handling, including modifying, fast search, extensive calculation facilities and more. Price £25.00 inclusive.
Model A or B.



SPACE WARP

The ultimate Star Trek type game, extremely complex, with high resolution colour graphics and sound. It is supplied complete with a sixteen page manual and function key labels. It requires 32K RAM. Price £11.50 inclusive. Suitable for the Model A(32K RAM) and the Model B.

ACTUAL SCREEN PHOTOGRAPH

THE BEEBON

A new magazine dedicated exclusively to users of the B.B.C. Micro. The Beebon contains tested programs, features on programming, special offers, features on hardware, reviews and much more. There will be at least three substantial programs in each issue, written by professional programmers. Published every two months, starting in May. Annual subscription (6 issues) only £7.50.

BACKGAMMON

A standard Backgammon game, for the Model A(32K RAM) and the Model B B.B.C. Micros . It has extremely fast computer resposes. Full playing instructions are included. Price £8.00.

PLEASE SUPPLY. . . . .
NAME: . . . .
ADDRESS: . . . .
I ENCLOSE A CHEQUE/POSTAL ORDER . . . .
OR PLEASE DEBIT MY ACCESS/BARCLAYCARD NUMBER. . . . .
SIGNATURE. . . . .
SEND TO: BUG-BYTE, 100 THE ALBANY, OLD HALL STREET, LIVERPOOL L3 9EP.

CONTENTS

Editorial & Advertisement Office 145 Charing Cross
Road, London WC2H 0EE
Telephone: 01-437 1002. Telex: 8811896

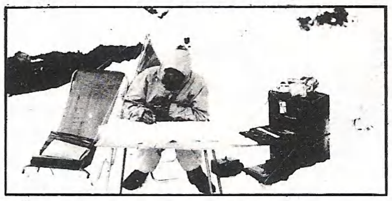
Preface 4

Technical

Morse Trainer 6
The key to success?
Base Changer 9
Fluent number translations

Games

Surround 12
Wrapping it up!
Patterns 15
The shape of things to come?
Crystal Ball 18
It's magic!
Breakthrough 21
The wall crumbles
Wordsquare 24
Find the buried words
Squish 28
Squash with a twist
Battle Of Britain 30
The Few fight once more
Fox & Hounds 34
Tally Ho!
The White Barrows 37
Adventures with logic
Ski-Run 42
Can you negotiate the poles?
Alien Attackers 45
Variations on a theme
Conquering Everest 49
Strategic summit scaling



Dambuster 54
Be a bouncing bomber
Leapfrog 57
Games theory explained
Obstacle Course 62
Simply amazing!

Financial

Investment Checker 66
Savings on show
Home Finance 68
Keeping track
of budgets

BARCLAYCARD
VISA
BARCLAYS BANK

Meter Minder 72
It's a gas . . .

Household

Plot 'n' Planner 76
No waste here
Calendar 78
Short of a date?

Educational

Descriptive Statistics 82
The title says it all
Newton's Laws 86
Program for stargazers
Root Solver 92
Complexities unravelled

Computing Today 20
Book your monthly copy here
ZX Computing 33
Our new quarterly for the
Sinclair user
User Groups 64
Independent information
Personal Software 74
Copies by post, post free!
Bibliography 95
Further reading
CT Software 96
Get them on tape
The Valley 98
Now available for BBC users

Credits: Our grateful thanks to the following individuals and organisations for their assistance in the production of this issue. Acorn Computers, BBC Publications, Centronics Data Computer, Hughes Aircraft Corporation, Microvitec, Paramount Pictures, Rank Film Distributors, Thorn-EMI Video Programmes Ltd, 20th Century Fox, Kieran, Peter and last, but by no means least, Sue, Lesley and Jane.

Personal
SOFTWARE
SUMMER 1982 £1.95
A COMPRHENSIVE library of tested and documented software for the BBC Microcomputer
Put your BBC micro to good use.
• Technical • Games • Financial
• Household • Educational
From the publishers of Computing Today

Volume 1
No 1
Summer 1982

Editor: Mike James
Advertisement Manager:
Coleen Pimm
Assistant Editor:
Roger Munford
Editorial Assistant:
Tina Boylan
Advertisement Copy Control: Sonia Hunt
Consultant Editor:
Henry Budgett
Managing Editor:
Ron Harris BSc
Managing Director: T J Connell

Origination and design by MM Design & Print

Personal Software is a quarterly magazine appearing on the third Friday in May, August, November and February.

Distribution by: Argus Press Sales & Distribution Ltd, 12-18 Paul Street, London EC2A 4JS Tel: 01-247 8233. Printing by: Alabaster Passmore & Sons Ltd, Tovil, Maidstone, Kent.

The contents of this publication including all articles, designs, plans, drawings and programs and all copyright and other intellectual property rights therein belong to Argus Specialist Publications Limited. All rights conferred by the Law of Copyright and other intellectual property rights and by virtue of international copyright conventions are specifically reserved to Argus Specialist Publications Limited and any reproduction requires the prior written consent of the Company. © 1982 Argus Specialist Publications Limited.

Subscription Rates : UK £7.80 for four issues including postage. Airmail and other rates upon application to Personal Software Subscriptions, 513 London Road, Thornton Heath, Surrey CR4 6AR.

PREFACE

So you have, or have access to, a BBC Micro. It's a super machine — it's got colour, it's got sound, it's got lots of different modes and it's really exciting to use. But it's also new and simply because it's so new, there's hardly any software to put it through its paces. The WELCOME (as shown on the cover) pack is good fun — for the first few days anyway — but there is a limit to how often you can run even the best of programs!

The BBC Micro runs a fairly standard BASIC which means that programs written for other machines using a fairly standard BASIC should run on it without a lot of trouble — well that's the theory anyway. In practice though it's a bit more tricky. Which is why, to save you hours of frustration, we've gone to the trouble of collecting together a selection of programs and converted them, or re-written them, so that they run incorporating features which make the best use of the BBC Micro's extras — its colour, its sound and so on.

Along the way I'll explain details of what to look out for in converting programs originally written for other micros — hints to help you if you want to try your hand at implementing your own favourites.

Six of the programs in the book have been written from scratch for the BBC Micro. Compare them with the others and you'll notice their structure is very different, much closer to the ideal of 'top down structured programming' the computer academics are always recommending to us. Well, in the case of the BBC Micro their advice is well worth following. It makes structured programming really straightforward and the rewards are easy-to-understand and easy-to-modify programs.

Each program, converted or original, is presented in a standard format. There is a general introduction, an outline of the program's structure (apart from the few cases where the programs are exceptionally straightforward), a technical discussion, a 'how to use it' section and, where appropriate, a discussion of what things can be done to take the program further. It should be possible to learn a great deal about how the program, and the BBC Micro, can be used by reading the program structure and technical sections.

To make it easier to enter the programs from their listings, each one has been renumbered in steps of 10. This means you can use the AUTO command to avoid having to type hundreds of line numbers. The renumbering was done after each program had been tested and the descriptions of the program structure are all based on this new numbering.

If you don't fancy typing the programs in from the listings, there is an easy option. CT Software have all the programs you'll see in these pages already on tape — for further details and ordering information, see the advertisement elsewhere in this magazine.

Technical

A	·—	S	···
B	—···	T	—
C	—·—·	U	··—
D	—··	V	···—
E	·	W	·——
F	··—·	X	—··—
G	——·	Y	—·——
H	····	Z	——··
I	··	1	·————
J	·———	2	··———
K	—·—	3	···——
L	·—··	4	····—
M	——	5	··
N	—·	6	—
O	———	7	—
P	·——·	8	—
Q	——·—	9	—

Morse Trainer — Practise Morse Code to perfection with this program. It will help you improve receiving speeds and aid initial recognition of Morse characters, introducing you to Morse Code gradually through a series of planned lessons.

4K

••—• •—• ——— ——

— •••• •

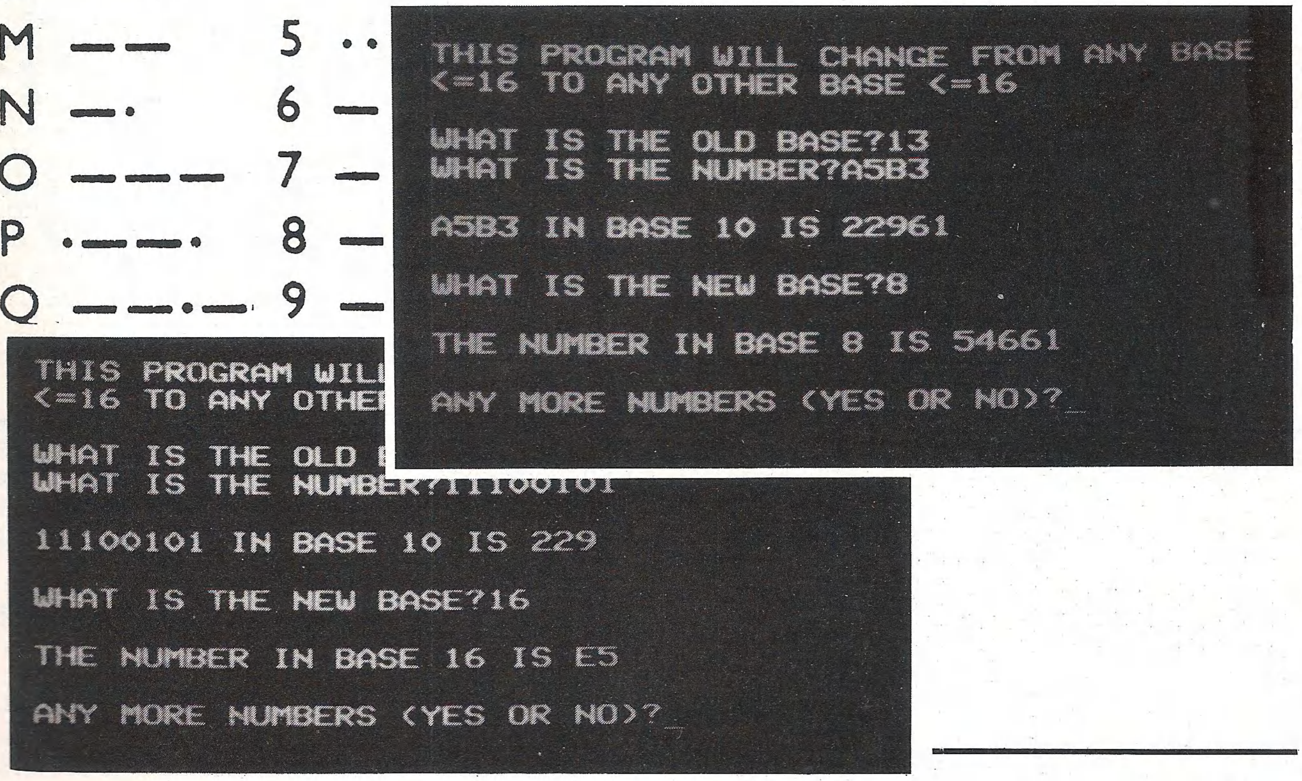
•——• ••— —•••

•—•• •• ••• •••••

•—• •••

——— ••—•

—•—• —



```
THIS PROGRAM WILL CHANGE FROM ANY BASE
<=16 TO ANY OTHER BASE <=16

WHAT IS THE OLD BASE?13
WHAT IS THE NUMBER?A5B3

A5B3 IN BASE 10 IS 22961

WHAT IS THE NEW BASE?8

THE NUMBER IN BASE 8 IS 54661

ANY MORE NUMBERS (YES OR NO)?_
```

```
THIS PROGRAM WILL
<=16 TO ANY OTHER

WHAT IS THE OLD
WHAT IS THE NUMBER?11100101

11100101 IN BASE 10 IS 229

WHAT IS THE NEW BASE?16

THE NUMBER IN BASE 16 IS E5

ANY MORE NUMBERS (YES OR NO)?_
```

Base Changer — Do you have trouble converting to and from binary or hexadecimal? This program takes away the guesswork, converting to and from any number base up to 16.

1.5K

Converted from a program by M K Cook

MORSE TRAINER

Confusing your dits and your dahs? Practise Morse code to perfection with this program.

A	·—	S	···
B	—···	T	—
C	—·—·	U	··—
D	—··	V	···—
E	·	W	·——
F	··—·	X	—··—
G	——·	Y	—·——
H	····	Z	——··
I	··	1	·————
J	·———	2	··———
K	—·—	3	···——
L	·—··	4	····—
M	——	5	·····
N	—·	6	—····
O	———	7	——···
P	·——·	8	———··
Q	——·—	9	————·
R	·—·	0	—————

Many people today are still interested in using Morse code. And well they might be, since even with all the technological advances which have been made, the human ear is still supreme in decoding a weak, interference-laden signal and the audio filtering between our ears remains the best available by several orders of magnitude. However, as the back of the head is not fitted with an output socket to take advantage of these filters, the decoding also has to be done in the brain!

Unfortunately this requires learning and usually involves listening to records, tapes or having friends send Morse code which you subsequently verify that you have taken down correctly. The trouble with this is that you soon become familiar with the contents of the records and tapes, and quickly lose your friends! The answer lies with your friendly computer, which will sit for hours contentedly churning out Morse code for you to learn without a single grumble.

This program can help to improve Morse code receiving speeds and aid the initial recognition of Morse characters. It introduces you to Morse code gradually in a series of lessons. In the first one, the letters are either all dots or all dashes and therefore easy to learn. As you progress, other letters are added but you are still tested on those you should already know.

A Load Of Code

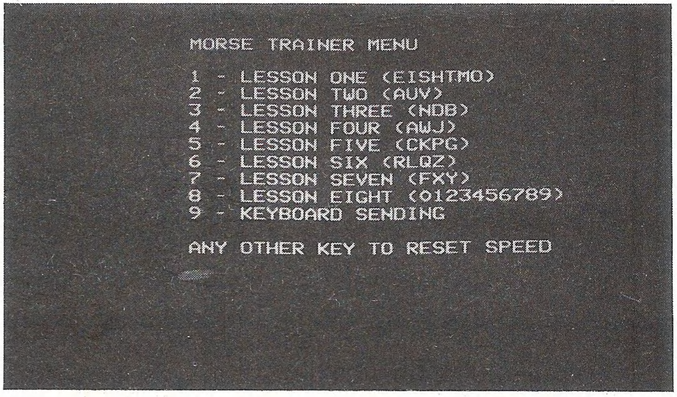
The Class A Radio Amateur Exam requires the sending and receiving of Morse at 12 words per minute, so that is the top speed for which this program is designed although higher speeds can be achieved by altering the appropriate delay values. In accord with the recommendations of the Radio Society of Great Britain, the dots and dashes are sent at 12 wpm even at the slower speeds — the spaces being increased to provide the lower overall speed. In this way the code is learned more easily. The program also provides a sound output via the cassette output port.

The secret of learning Morse code is regular practise every day for 15 to 30 minutes. You should start with the slowest speed and work right through the lessons again — do not learn lesson one at 12 wpm before proceeding to lesson two.

TECHNICAL DETAILS

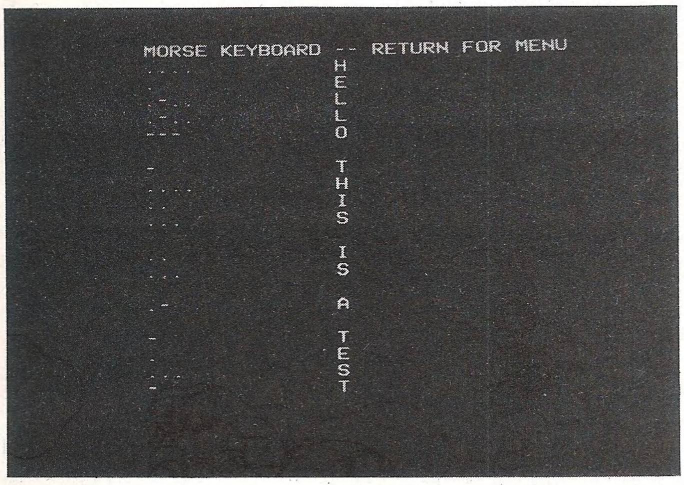
L is an array which contains the binary equivalent of the Morse code character in an array position corresponding to its ASCII value. For example, L(65) contains 6 (the Morse value of A). This is calculated by taking the Morse code for A

Right: The eight training lessons offered by the program are supplemented by a keyboard sender.



```
MORSE TRAINER MENU

1 - LESSON ONE (EISHTMO)
2 - LESSON TWO (AUV)
3 - LESSON THREE (NDB)
4 - LESSON FOUR (AWJ)
5 - LESSON FIVE (CKPG)
6 - LESSON SIX (RLQZ)
7 - LESSON SEVEN (FXY)
8 - LESSON EIGHT (0123456789)
9 - KEYBOARD SENDING

ANY OTHER KEY TO RESET SPEED
```

Left: An example of the keyboard sender in operation. You type the message and the program sends the Morse.



```
MORSE KEYBOARD -- RETURN FOR MENU
                  H
                  E
                  L
                  L
                  O

                  T
                  H
                  I
                  S

                  I
                  S

                  A

                  T
                  E
                  S
                  T
```

which is '.-'. This is then written in the reverse order, using a 0 for a dot and a 1 for a dash, thus 10. A 1 is placed on the left-hand side of this number to indicate that the character is complete. The resulting number (110) is then converted from binary and becomes 6.

To output the Morse, the least significant bit is tested; for 0 a dot is produced and for a 1, a dash. The number is then shifted one place to the right and the process is repeated until the number is 1, indicating that the entire character has been sent. The duration of the tone is set in line 90 and the inter-element space in variable DE.

HOW TO USE THE PROGRAM

Each lesson can be one of two types. First, there is keyboard recognition: in this mode the computer gives you a Morse character which you have to match on the keyboard. If you get it right, the computer tells you so and one point is added to your score and to your 'tries' total; if you get it wrong you will again be told, just one point being added to your tries total. You will keep getting the same character until you get it right.

Once you can obtain a good score on that, go to the second mode in which five letter random groups are given. Write these down without looking at the screen and check them when the group is completed.

Incidentally, the characters appear on the screen just after the sound, so a beginner could guess in advance of the characters and subconsciously learn Morse.

▶

VARIABLES

L(96)	Array with Morse code for conversion.
SP(4)	Array with inter-letter space delay.
SW(4)	Array with inter-word space delay.
LS$	String variable with the current lesson's characters.
M	Morse characters used by output routine.
GR	Variable set to 0 if dots and dashes are to be displayed graphically, otherwise set to 1.

PROGRAM STRUCTURE

Statement	Function	Action
Lines 10-80	Set Up	Selects MODE 7 and defines arrays used.
Lines 90-200	Speed	User is asked to select speed for practise.
Lines 210-380	Menu	Choice of the nine lessons displayed. Goes to appropriate section according to user's response.
Lines 390-450	Morse Keyboard	Sends letter typed on keyboard.
Lines 460-630	Preparation	Sets up the values of the letters for lessons.
Lines 640-850	Lesson Content	General lesson routine — displays content of forthcoming lesson.
Lines 860-1380	Lesson	User given option of two types of practise and fuller instructions. Computer then gives required signals, checks responses and gives appropriate feedback.
Lines 1390-1430	Dot	Routine to print dot as '.' and to give sound of appropriate length.
Lines 1440-1490	Dash	Routine to print dash as '-' and to give sound of appropriate length.
Lines 1500-1560	Timings	Data arrays for timing.

```
 10   MODE 7
 20   DIM L(96),SP(4),SW(4)
 30   FOR A=32 TO 95
 40   READ L(A)
 50   NEXT A
 60   FOR A=1 TO 4
 70   READ SP(A),SW(A)
 80   NEXT A
 90   TS=2:TL=TS*3:REM**TIME OF TONE
100   CLS:PRINT:PRINT TAB(6);
      " ** MORSE TRAINER **"
110   PRINT:PRINT:PRINT:PRINT"SELECT
      SPEED ---------- "
120   PRINT:PRINT TAB(10);"1 = 6  WORDS/MIN"
130   PRINT TAB(10);"2 = 8  WORDS/MIN"
140   PRINT TAB(10);"3 = 10 WORDS/MIN"
150   PRINT TAB(10);"4 = 12 WORDS/MIN"
160   K$=GET$:IF K$="" THEN 160
170   T=ASC(K$)
180   IF T>52 OR T<49 THEN PRINT"PLEASE
      ENTER 1 TO 4":GOTO 160
190   DE=SP(VAL(K$))
200   WD=SW(VAL(K$))
210   CLS:PRINT:PRINT "MORSE TRAINER MENU"
220   GR=0:NW=0:R=0
230   PRINT
240   PRINT "1 - LESSON ONE (EISHTMO)"
250   PRINT"2 - LESSON TWO (AUV)"
260   PRINT"3 - LESSON THREE (NDB)"
270   PRINT"4 - LESSON FOUR (AWJ)"
280   PRINT"5 - LESSON FIVE (CKPG)"
290   PRINT"6 - LESSON SIX (RLQZ)"
300   PRINT"7 - LESSON SEVEN (FXY)"
310   PRINT"8 - LESSON EIGHT (0123456789)"
320   PRINT"9 - KEYBOARD SENDING"
330   PRINT
340   PRINT"ANY OTHER KEY TO RESET SPEED"
350   K$=GET$:IF K$="" THEN 350
360   IF ASC(K$)>57 OR ASC(K$)<49 THEN 100
370   SW=0:LS$=""
380   ON VAL(K$) GOTO 610,590,570,550,
      530,510,490,460,390
390   CLS:PRINT:PRINT "MORSE KEYBOARD
      -- RETURN FOR MENU"
400   K$=GET$:IF K$="" THEN 400
410   IF K$=CHR$(13) THEN 210
420   M=L(ASC(K$))
430   IF M=0 THEN GOSUB 1470 ELSE GOSUB 1320
440   PRINT TAB(15);K$
450   GOTO 400
460   LS$="0123456789"
470   GOSUB 640
480   GOTO 860
490   LS$="FXY"
500   GOSUB 640
510   LS$=LS$+"RLQZ"
520   IF NW=0 THEN GOSUB 640
530   LS$=LS$+"CKPG"
540   IF NW=0 THEN GOSUB 640
550   LS$=LS$+"AWJ"
560   IF NW=0 THEN GOSUB 640
570   LS$=LS$+"NDB"
580   IF NW=0 THEN GOSUB 640
590   LS$=LS$+"AUV"
600   IF NW=0 THEN GOSUB 640
610   LS$=LS$+"EISHTMO"
620   IF NW=0 THEN GOSUB 640
630   IF R=1 THEN 210 ELSE 860
640   CLS:PRINT:PRINT "LESSON " ;VAL(K$);
      " - IN WHICH WE LEARN"
650   PRINT
660   IF K$="8" THEN PRINT"NUMBERS    ";
      ELSE PRINT "LETTERS     ";
670   FOR A=1 TO LEN(LS$)
680   PRINT MID$(LS$,A,1);" ";
690   NEXT A
700   PRINT:PRINT
710   FOR A=1 TO 500:NEXT A
720   GR=0
730   FOR A=1 TO LEN(LS$)
740   K$=MID$(LS$,A,1)
750   M=L(ASC(K$))
760   GOSUB 1320
770   PRINT TAB(11);K$
780   GOSUB 1470
790   NEXT A
800   PRINT
810   PRINT"PRESS 'R' TO REPEAT"
820   PRINT"ANY KEY TO CONTINUE"
830   K$=GET$:IF K$=""THEN 830 ELSE  IF
      K$="R" THEN 730
840   IF K$=CHR$(13) THEN R=1
850   NW=1:RETURN
860   CLS:PRINT:PRINT "PLEASE SELECT AN OPTION"
870   PRINT
880   PRINT" 1 - KEYBOARD RECOGNITION "
890   PRINT"      I GIVE A SOUND AND YOU"
900   PRINT"      MATCH IT ON THE KEYBOARD"
910   PRINT:PRINT" 2 - RANDOM GROUPS"
920   PRINT"      I GIVE YOU 12 GROUPS AND"
930   PRINT"     TAKE THEM DOWN WITHOUT LOOKING"
940   K$=GET$:IF K$="" THEN 940
950   IF K$="2" THEN 1150
960   IF K$<>"1" THEN 210
970   CLS:PRINT:PRINT "ECHO EACH
      CHARACTER  WHEN YOU HEAR  IT"
980   PRINT:PRINT "PRESS RETURN TO
      SELECT AGAIN"
990   PRINT:PRINT:PRINT TAB(12);"*** SCORE ***"
1000  PRINT TAB(8);"CORRECT";SPC(7);"OUT OF"
1010  FT=0:AT=0
1020  GR=1:REM**TURN OF GRAPHICS
1030  K1$=MID$(LS$,RND(LEN(LS$)),1)
1040  M=L(ASC(K1$))
1050  PRINT TAB(8,8); FT;TAB(15);AT
1060  GOSUB 1470
1070  PRINT
1080  GOSUB 1320
1090  K$=GET$:IF K$=""THEN 1090
1100  IF K$=CHR$(13) THEN 860
1110  PRINT K$
1120  IF K$<>K1$ THEN AT=AT+1:PRINT"*** NO TRY
      AGAIN ***":GOTO 1040 ELSE PRINT" -OK- ";
      SPC(15)
1130  FT=FT+1:AT=1+AT
1140  GOTO 1030
1150  CLS:GR=1:PRINT
1160  PRINT"RANDOM GROUPS"
1170  PRINT
1180  FOR A=1 TO 1000:NEXT A
1190  FOR A=1 TO 2
1200  FOR AW=1 TO 5
1210  K$=MID$(LS$,RND(LEN(LS$)),1)
1220  M=L(ASC(K$))
1230  GOSUB 1320
1240  PRINT" ";K$;
1250  NEXT
1260  PRINT
1270  GOSUB 1470
1280  NEXT
1290  PRINT:PRINT"HIT RETURN FOR MENU
      ANY KEY TO REPEAT"
1300  K$=GET$:IF K$=""THEN1300
1310  IF K$=CHR$(13) THEN 860 ELSE 1150
1320  T=(M AND 1)+1
1330  ON T GOSUB  1390,1440
1340  M=INT(M/2)
1350  IF M<>1 THEN 1320
1360  FOR A1=1 TO DE
1370  NEXT A1
1380  RETURN
1390  IF GR=0  THEN PRINT ".";
1400  TIME=-TS*5:SOUND 1,-15,150,TS
1410  REM INTER LETTER DELAY
1420  IF TIME<DE THEN 1420
1430  RETURN
1440  IF GR=0 THEN PRINT "-";
1450  TIME=-TL*5:SOUND 1,-15,150,TL
1460  GOTO 1410
1470  TIME=0
1480  IF TIME<WD THEN 1480
1490  RETURN
1500  DATA 0,0,94,0,0,0,0,94,45,109,0,0,
      115,97,106,41
1510  DATA 63,62,60,56,48,32,33,35,39,47
1520  DATA 71,0,0,0,0,76,0
1530  DATA 6,17,21,9,2,20,11,16,4,30,13,
      18,7,5,15,22
1540  DATA 27,10,8,3,12,24,14,25,29,19
1550  DATA 0,0,0,0,0
1560  DATA 48,122,28,78,17,52,10,30
```

Converted from a program by W S Lounds

BASE CHANGER

Switching from decimal to binary and from binary to Hex is a chore that would be nice to hand over to the computer.

This program enables you to convert to and from any number base up to and including base sixteen, hexadecimal. It uses a novel method of extracting numerical information from alphanumeric data, relying on the ASCII code to do the work for you.

TECHNICAL DETAILS

The only clever part of this program is the way the ASCII code for each digit is converted to the numeric value. The input number string is broken down into digits using the MID$ function and then the number or letter is converted to numerical form by the ASC function. If the digit is in the range 0-9 then subtracting 48 from the ASCII code will give a number with the correct value and if the digit is a letter in the range A to F, subtracting 55 performs the same function. For example, if the number A9 is input, the first digit is the character 9 which has ASCII code 57, thus subtracting 48 gives the correct numeric value of 9.

This procedure seems to be

PROGRAM STRUCTURE

Statement	Function	Action
Lines 10-20	Set Up	Selects MODE 7 and clears screen.
Lines 30-70	Initial Base	Displays instructions and asks for base to convert from.
Lines 80-190	Get Number	Asks for number to be input and converts it to base 10.
Lines 200-230	Range Check	Checks whether number is less than a million; if not displays warning.
Lines 240-260	Final Base	Asks for the base into which number is to be converted.
Lines 270-350	Conversion	Converts number to appropriate base.
Lines 360-390	Answer	Displays answer.
Lines 400-430	Repeat Loop	Allows option of performing another conversion.
Lines 440-470	String Conversion	Converts from string to number.
Lines 480-560	Base 10 Conversion	Converts to base 10.

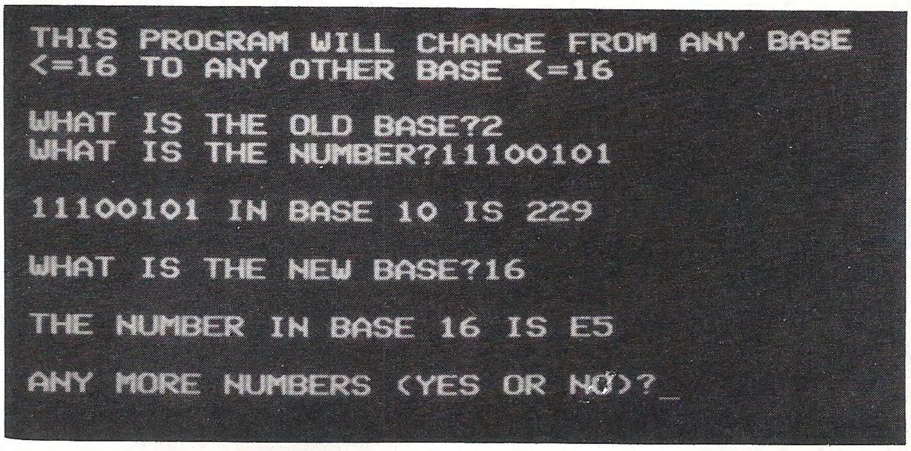
```
 10  MODE 7
 20  CLS
 30  PRINT
 40  PRINT "THIS PROGRAM WILL CHANGE
     FROM ANY BASE"
 50  PRINT "<=16 TO ANY OTHER BASE <=16"
 60  PRINT
 70  INPUT "WHAT IS THE OLD BASE",X$
 80  E=0
 90  IF X$=""THEN 50
100  GOSUB 440
110  B=N
120  IF N<2 OR N>16 THEN 50
130  INPUT "WHAT IS THE NUMBER",X$
140  IF X$=""THEN 130
150  GOSUB 480
160  IF E=1 THEN PRINT " ERROR ":E=0:GOTO 130
170  N1=N
180  PRINT
190  PRINT X$;" IN BASE 10 IS ";N1
200  IF N1<1000000 THEN 240
210  PRINT "THE NUMBER IN BASE 10 IS
     >=1000000, THIS"
220  PRINT "MEANS THAT ERRORS MAY OCCUR"
230  PRINT
240  PRINT:INPUT "WHAT IS THE NEW BASE",X$
250  IF X$="" THEN 240
260  GOSUB 440
270  B1=N:IF N<2 OR N>16 THEN 240
280  B$=""
290  V=INT(N1/B1)
300  R=N1-V*B1
310  IF R>9 THEN 350
320  B$=B$+CHR$(R+48)
330  N1=V:IF V=0 THEN 360
340  GOTO 290
350  R=R+55:B$=B$+CHR$(R):N1=V:
     IF V<>0 THEN 290
360  PRINT: PRINT "THE NUMBER IN BASE ";
     B1;" IS ";
370  FOR J=LEN(B$) TO 1 STEP -1
380  PRINT MID$(B$,J,1);:NEXT
390  PRINT
400  PRINT:INPUT "ANY MORE NUMBERS (YES
     OR NO)",X$
410  IF LEFT$(X$,1)="Y" THEN 20
420  IF LEFT$(X$,1)="NO" THEN STOP
430  GOTO 400
440  N=0
450  FOR J=1 TO LEN(X$):D=ASC(MID$(X$,J,1))
460  N=N*10+D-48:NEXT
470  RETURN
480  N=0
490  FOR J=1 TO LEN(X$):D=ASC(MID$(X$,J,1))
500  IF D>47 AND D<58 THEN D=D-48:GOTO 530
510  IF D>64 AND D<71 THEN D=D-55:GOTO 530
520  E=1:RETURN
530  IF D>=B THEN E=1:RETURN
540  N=N*B+D
550  NEXT J
560  RETURN
```

a little more useful when the second digit is converted. The second digit is the character A and has the ASCII value of 65. Subtracting 55 gives the correct numeric value of 10.

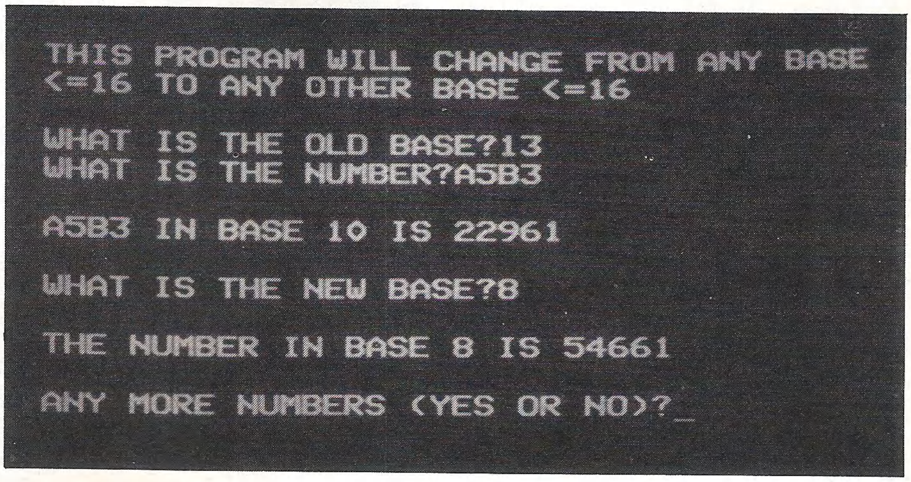
HOW TO USE THE PROGRAM

First, the computer asks you to specify the base you wish to convert from — the 'old' base. It then asks for the number; it will accept any number but if the number is greater than 1,000,000 it prints a warning that errors may occur. It then asks you to specify the base you wish to convert to (the 'new' base') and gives you your number in that base.

Having answered one problem it asks 'Any more numbers' and if you reply 'YES', it starts the process from scratch.



```
THIS PROGRAM WILL CHANGE FROM ANY BASE
<=16 TO ANY OTHER BASE <=16

WHAT IS THE OLD BASE?2
WHAT IS THE NUMBER?11100101

11100101 IN BASE 10 IS 229

WHAT IS THE NEW BASE?16

THE NUMBER IN BASE 16 IS E5

ANY MORE NUMBERS (YES OR NO)?_
```



```
THIS PROGRAM WILL CHANGE FROM ANY BASE
<=16 TO ANY OTHER BASE <=16

WHAT IS THE OLD BASE?13
WHAT IS THE NUMBER?A5B3

A5B3 IN BASE 10 IS 22961

WHAT IS THE NEW BASE?8

THE NUMBER IN BASE 8 IS 54661

ANY MORE NUMBERS (YES OR NO)?_
```

Top: A simple binary to Hex conversion — just the sort of routine job the program was designed for.

Right: Could you convert base 13 into Octal in your head?!

MICROCOMPUTER COMPONENTS
LOWEST PRICES - FASTEST DELIVERY

MEMORIES AT UNBEATABLE PRICES

		1+	25+	100+
2114 Low Power	200ns	0.93	0.89	0.84
2708	450ns	2.25	1.99	1.80
2716	450ns	2.45	2.25	1.95
2532	450ns	4.50	4.25	4.05
2732	450ns	4.95	4.65	4.20
4116	150ns	0.93	0.89	0.84
4116	200ns	0.74	0.70	0.67
4118	200ns	3.90	3.45	2.99
5516	200ns	9.50	8.95	8.50
6116	200ns	5.70	5.35	5.05
6116 Low Power	200ns	7.60	7.20	6.75

*Official orders from educational establishments qualify for 100+ price.

OFFICIAL ORDERS WELCOME
VISA
24 HOUR TELEPHONE SERVICE FOR CREDIT CARD USERS
QUANTITY DISCOUNTS AVAILABLE

PLEASE SEND S.A.E (20p) FOR OUR NEW 1982 CATALOGUE. FREE REPLIED PAID ENVOLOPE WITH EVERY ORDER. ALL PRICES EXCLUDE P & P ON ORDERS UNDER £10 (75p) OVER £10 CASH WITH ORDER AT LIST PRICE ONLY AND VAT (15%). ALL ORDERS DESPATCHED ON DAY OF RECEIPT WITH FULL REFUND FOR OUT OF STOCK ITEMS IF REQUESTED.

MIDWICH COMPUTER CO. LTD.
DEPT CT, HEWITT HOUSE, NORTHGATE STREET,
BURY ST. EDMUNDS, SUFFOLK IP33 1HQ
TELEPHONE: (0284) 701321 TELEX: 817670

REGISTERED REFERRAL CENTRE
FOR THE BBC PROJECT

BEEBUG FOR THE BBC MICRO

INDEPENDENT NATIONAL USER GROUP FOR THE BBC MICRO

IF YOU OWN A BBC MACHINE, OR HAVE ORDERED ONE, OR ARE JUST THINKING ABOUT GETTING ONE, THEN YOU NEED BEEBUG.
BEEBUG runs a regular magazine devoted exclusivly to the BBC Micro (10 issues per year).

New program listings in each issue:
3.D. Noughts and Crosses and **Full Colour Moon Lander** in April issue.
Latest news on the BBC project.
What you should know before you order your machine.
Hardware and Software Hints and Tips: **What the Manual doesn't tell you.**
How to upgrade the A option.
How to add joysticks and games paddles.
Reviews of the latest software.
Discount software and hardware.
Regular advice Clinic to answer your queries.

All this and much more when you join BEEBUG.

MEMBERSHIP:
Introductory offer (closes 30th June)
6 months £4.50
1 year £8.50
After 30th June
£4.90 & £8.90
Send £1.00 & A S.A.E. for sample copy.

Make cheques payable to
BEEBUG
DEPT 10
374 WANDSWORTH RD.
LONDON, SW8 4TE

For technical enquiries and editorial material Send to:
The Editor, P.O. Box 50, St. Albans, Herts.

Games

Surround — The object is to move your piece around a moving 'enemy' so that he has nowhere to go! Fast reactions are needed for this one. **2.5K**

Patterns — A demonstration not only of the BBC Micro's graphics capabilities but also of how restful moving colour images can be. Several choices of pattern including some fantastic web designs which continuously alter. **1.5K**

Crystal Ball — Type in a name and get the computer to generate a lighthearted character assessment based on a reduction of the subject's name to a single 'magic' number. **3.5K**

Breakthrough — You have a bat, five balls and a wall of six rows of bricks. The aim is to break through the barrier of bricks scoring as many points as possible. **2.5K**

Wordsquare — The idea is to give the computer a list of words which it then hides in a 15 by 15 grid of letters. They may be hidden in a number of different ways and no two puzzles are ever the same. **4K**

Squish — No game of squash was ever like this! Between you and the back wall are a row of pillars which the 'ball' can bounce off! **2K**

Battle of Britain — Based on two months at the height of the Battle of Britain, you allocate the number of British radar sites to be implemented, the number of fighters to scramble, the number of fighters to manufacture, etc. The computer plays the German side. **2.5K**

Fox and Hounds — A game played on a chess-board grid where the computer controls the fox and you control the hounds. The aim is to trap the fox. **3.5K**

The White Barrows — Trap the Sorcerer and recover the Magic Chalice, using your magical powers carefully to negotiate and locate his position. Bribe goblins, fight dragons and work out the layout of the labyrinth. **5.5K**

Ski-Run — Downhill slalom complete with gates to negotiate. Reach the bottom of the course in the shortest possible time, steering your way around the gates and without going off the edge of the course. **3K**

Alien Attackers — Control the position of your space ship so that you can shoot down the invading aliens. They seem to travel slowly but they never stop! **3K**

Conquering Everest — Have you ever dreamt of climbing our world's highest mountain? This impressive text-only game allows you to travel from base to base, choosing whether you want porters or Sherpas to carry all your provisions. But don't expect to conquer Everest at one sitting, you have to move the right equipment to the right place at just the right time. **8.5K**

Dambuster — Watch those bombs bounce as they hit the water . . .will you destroy the dam this time? **2.5K**

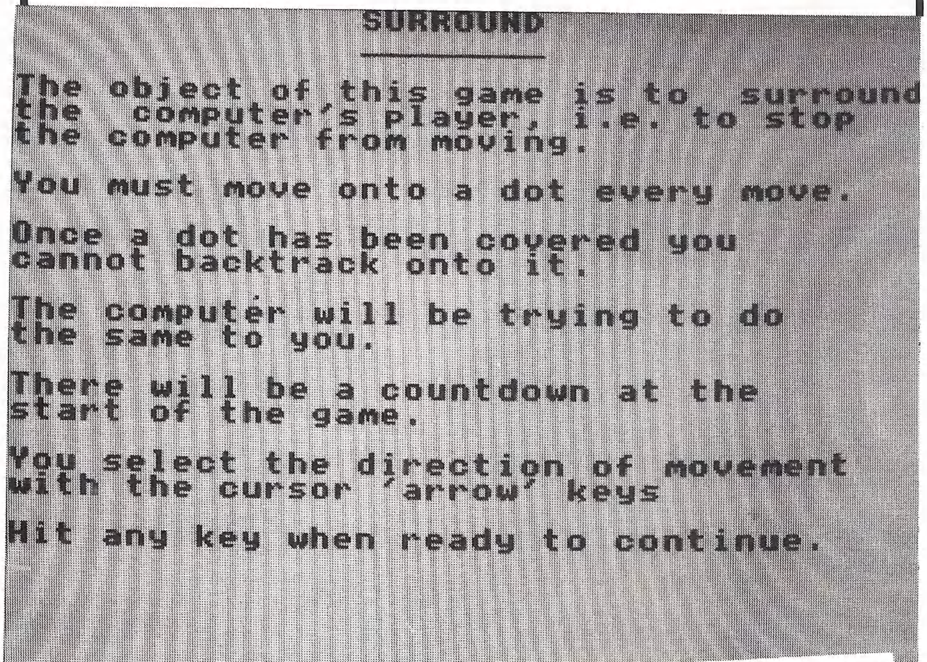
Leapfrog — The aim of the game is to get the left-hand rows of 'coins' over to the right-hand side and vice versa. Sounds easy . . . **5.5K**

Obstacle Course — Race against the clock to get through a maze which is continually changing its pattern. Grit your teeth when it blocks you but don't despair, you'll get through in the end. **1.5K**

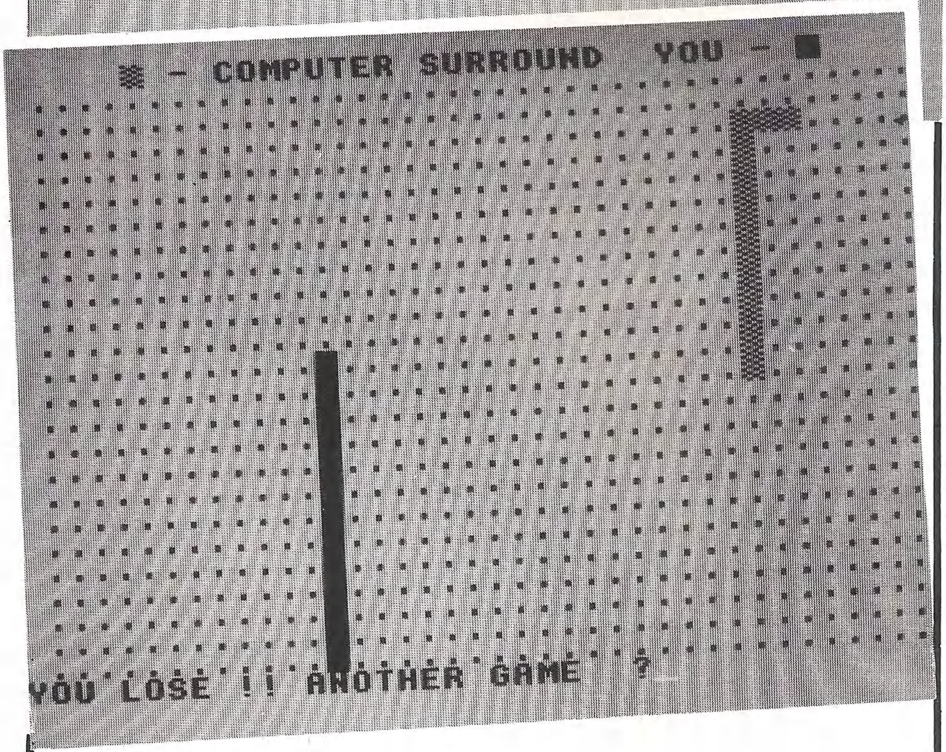
Converted from a program by Andrew Thomas

SURROUND

This program could easily give you that 'hemmed-in' feeling.



```
                SURROUND

The object of this game is to  surround
the  computer's player, i.e. to stop
the computer from moving.

You must move onto a dot every move.

Once a dot has been covered you
cannot backtrack onto it.

The computer will be trying to do
the same to you.

There will be a countdown at the
start of the game.

You select the direction of movement
with the cursor 'arrow' keys

Hit any key when ready to continue.
```



```
   ▓ - COMPUTER SURROUND  YOU - ■

YOU LOSE !! ANOTHER GAME  ?
```

The pictures above and on the opposite page show the initial display and two examples of the game in action. In the first game display, the player has lost by attempting to retrace his own steps, in the second the player has tried to cross his previous tracks.

Surround is well-known as a classic PET game but this conversion allows for an equally entertaining game on the BBC Micro.

At the beginning of the game the screen fills with a grid of dots and two squares appear. One is yours, the other belongs to your challenger — the computer. The object of the game is to surround your opponent so that while you try to hem in the computer, the computer is attempting to surround you. You have just one advantage over the machine — you can select the skill level at which you want to play. This selection determines the speed at which the program will run.

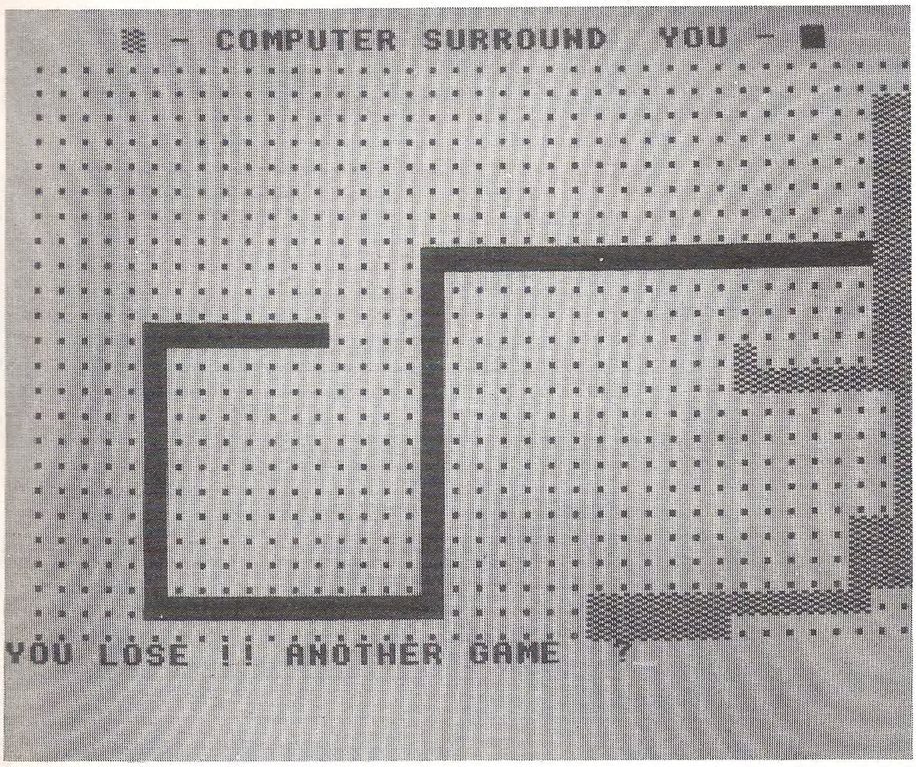
Note once the skill level is set, the only way to alter it is to Escape and re-RUN the program.

TECHNICAL DETAILS

Two methods are used to convert this PET program for the BBC Micro. The original program made use of the fact that the PET's display is memory-mapped; it placed graphics characters on the screen by POKING memory locations and found out what was on the screen by PEEKing memory locations. The BBC Micro is also a memory mapped system but it works in such a totally different way that simply changing the PEEK or POKE addresses is not enough.

The easier of the two problems to solve is the POKEing of characters onto the screen. In MODE 4 the BBC Micro has a 40 by 32 line screen. The PET's screen is only 40 by 25 lines so it should be possible to fit any PET display into a MODE 4 screen.

The first step is to convert the PET screen address into x,y co-ordinates corresponding to the same point on the screen. In Surround, procedure PROCXY(M) converts the PET's



▒ - COMPUTER SURROUND YOU - ■

YOU LOSE !! ANOTHER GAME ?

PROGRAM STRUCTURE

Statement	Function	Action
Lines 10-60	Set Up	Selects graphics mode, defines CHR$(224) as a stippled square, CHR$(225) as a solid square and CHR$(226) as a square with its left-hand side solid and its right-hand side blank. The foreground and background colours are set to yellow and black. The arrow keys are brought into use.
Lines 70-320	Instructions	Displays information. Selects skill level.
Lines 330-410	Initialise	Initialises variables.
Lines 420-490	Print Board	Displays information and selects skill places each opponent's starting square on the board.
Lines 500-540	Countdown	Displays ten blocks at bottom right of screen which vanish one by one. The disappearance of the final block signals start of play.
Lines 550-600	Get Key	Selects direction of movement.
Lines 610-730	Move Logic	Checks whether a move is allowed and whether either opponent has won.
Lines 740-790 840,880	End Game	Notes whether game is lost or won. Resets to normal mode (MODE 7) and restores arrow keys to usual function.
Lines 800-870	Set Values	Controls move values.
Lines 850-870 890	Address	Defines screen addresses

memory address M to x,y co-ordinates using the simple equations:

$$Y = INT((M-32768)/40)$$
$$X = M-32768-40*Y$$

These x,y co-ordinates are used by the BBC Micro in PRINT TAB(X,Y);CHR$(C); statements to print the character C at the same location as on the PET's screen. This technique is also used in many of the other programmes in this issue where conversion from a POKE-based system is needed.

The problem of producing the graphics character used by the PET is solved by using the VDU 23 call (this allows special characters to be defined). For Surround, three extra characters are required:

224, a dotted square

225, a solid square

226, a half square

In general, the PEEK problem is much more difficult to solve. You can convert the PET's PEEK addresses to x,y co-ordinates by using PROCXY and you can then go on and convert the x,y co-ordinates to BBC Micro screen addresses.

The trouble is that the BBC Micro doesn't store the ASCII code of the character at the screen location — it stores the actual dot pattern of the shape of the character in ***eight*** memory locations. Each character is made up of eight rows of eight dots and each row is stored in a memory location. This means that you can PEEK a screen location to find out what character is at any position but there is still the question of which one of the eight should you examine? It is obvious that the value that you get back depends on which row of the character you PEEK.

The solution to this problem is to choose to PEEK a row of the character that identifies it as different from all the other characters in use. For example, in Surround you can tell the difference between all the characters by looking at the first row or the seventh row. If ▶

character 224 is present at a screen position then PEEKing the address which corresponds to its first row returns a value of $CC. Similarly character 225 would return $FF and character 226 would return $F0.

The only other character used in Surround is a '.' and this returns the value 0 if you look at the top row. This is the same value you would get from the character 'blank'. To tell the difference between a blank and a '.' you have to look at the seventh row which contains the value $24.

It should be clear that you can tell which character, if any, is at any location simply from the value of the top row or the seventh row. The function FNS(X,Y,N) calculates the screen address of the Nth row of the character at x,y. The BBC Micro follows the tradition set by the ATOM in using '?' instead of the more usual PEEK. Thus the rather cryptic ?FNS(X,Y,N) returns the value of the Nth row of the character at screen position x,y.

Other features worth noting are the use of *FX 4,1 to bring the cursor control keys into the keyboard as 'normal' keys, the use of VDU 19 to set the background/foreground colours and the use of STRING$ to produce the playing area.

HOW TO PLAY

Using the cursor control arrows to move, you have to try to surround your opponent (the computer) so that it cannot make a legal move. You are not allowed to move backwards onto a square you have already covered, you must not leave the screen and you cannot move onto a square that the computer has already covered. If you make a prohibited move you automatically lose the game.

You can select the speed at which the game is played by choosing your 'skill factor'; level 9 gives the slowest and easiest game and level 1, the fastest and most difficult.

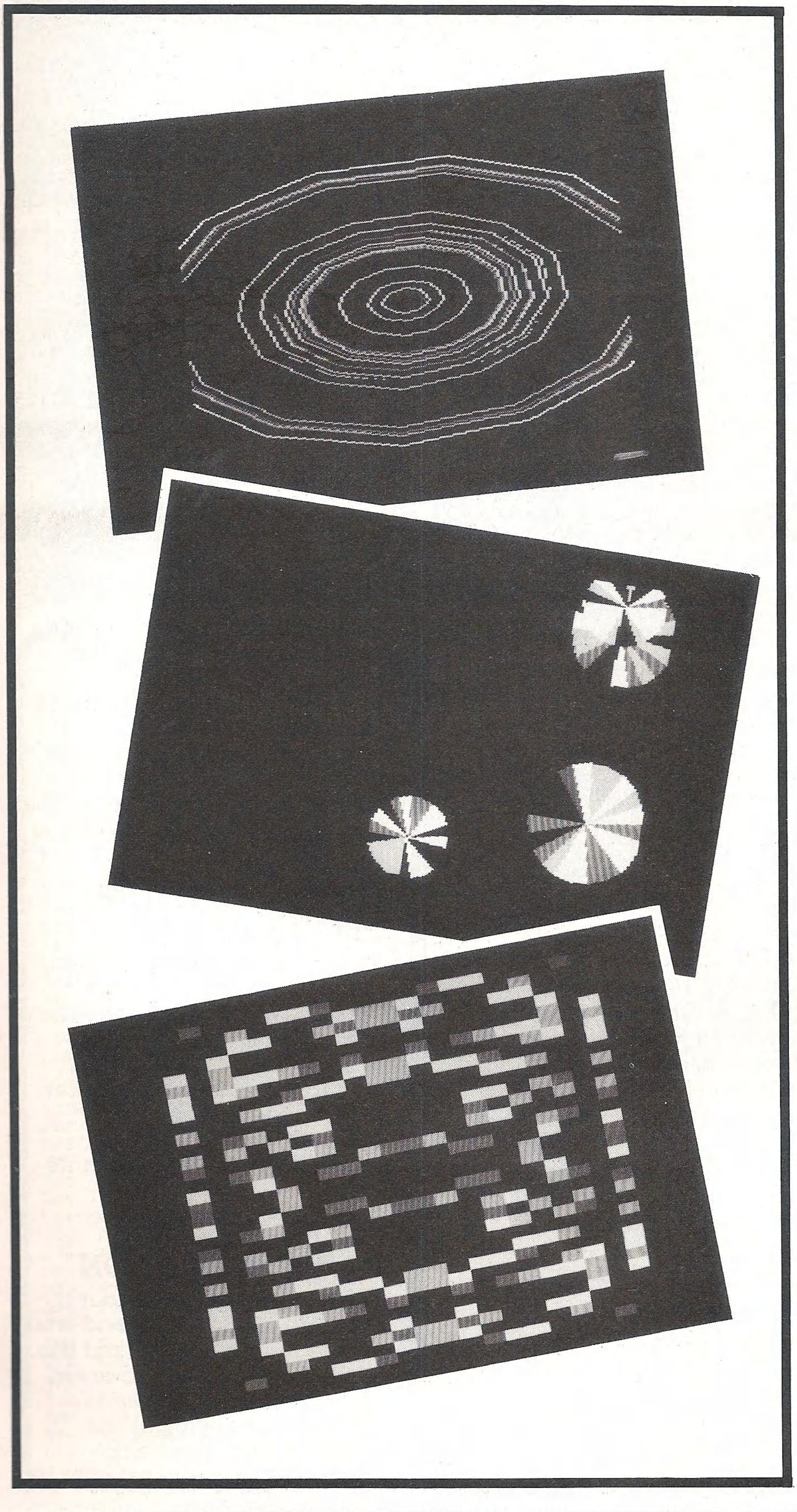
The start of the game is signalled by the 'countdown' in the bottom right-hand corner — as the last block disappears the computer is on its way.

```
 10  VDU 23,224,&CC,&33,&CC,&33,&CC,&33,&CC,&33
 20  VDU 23,225,&FF,&FF,&FF,&FF,&FF,&FF,&FF,&FF
 30  VDU 23,226,&F0,&F0,&F0,&F0,&F0,&F0,&F0,&F0
 40  *FX 4,1
 50  DIM A(4)
 60  MODE 4:CLS:1,0,0,0,0:
     VDU 19,0,3,0,0,0
 70  PRINT TAB(15); "SURROUND"
 80  PRINT TAB(15); "_________"
 90  PRINT
100  PRINT "The object of this game is
     to  surround"
110  PRINT "the  computer's player,
     i.e. to stop"
120  PRINT "the computer from moving."
130  PRINT
140  PRINT "You must move onto a dot
     every move."
150  PRINT
160  PRINT "Once a dot has been covered you"
170  PRINT "cannot backtrack onto it."
180  PRINT
190  PRINT "The computer will be trying
     to do"
200  PRINT "the same to you."
210  PRINT
220  PRINT "There will be a countdown
     at the"
230  PRINT "start of the game."
240  PRINT
250  PRINT"You select the directio  of
     movement"
260  PRINT"with the cursor 'arrow' keys"
270  PRINT
280  PRINT "Hit any key when ready to
     continue."
290  A$=GET$
300  PRINT
310  INPUT "Select your level between 1,
      difficult   and 9, easy ",S
320  IF S<1 OR S>9 THEN PRINT
     "I said ";:GOTO 310
330  S=S*100
340  A(1)=1
350  A(2)=-1
360  A(3)=40
370  A(4)=-40
380  X1=33359
390  Y1=A(RND(4))
400  Y=1
410  X=33301
420  CLS
430  PRINT
440  PRINT TAB(5); CHR$(224);
     " - COMPUTER SURROUND  YOU - ";CHR$(225)
450  FOR I=1 TO 24
460  PRINT SPC(1); STRING$(38,".")
470  NEXT I
480  PROCXY(X):PRINT TAB(XC,YC);CHR$(225);
490  PROCXY(X1):PRINT TAB(XC,YC);CHR$(224);
500  FOR I=0 TO 10
510  PROCXY(33847-I):PRINT TAB(XC,YC);
     CHR$(226);
520  NEXT I
530  FOR I=10 TO 0 STEP-1:FOR J=1 TO 500:
     NEXT J
540  PROCXY(33847-I):PRINT TAB(XC,YC);
     " ";:NEXT I
550  A=RND(4)
560  FOR I=1 TO S:NEXT I
570  I=INKEY(0)
580  IF I<>-1 THEN I=I-&87:A=I
590  *FX 15,0
600  ON A GOSUB 810,820,800,830
610  IF A<>0 AND RND(1)>.5 THEN Y1=-Y
620  X=X+Y
630  PROCXY(X):IF ?FNS(XC,YC,6)<>24 THEN 750
640  PRINT TAB(XC,YC);CHR$(225);
650  PROCXY(X1+3*Y1):
     IF ?FNS(XC,YC,6)<>24 THEN 690
660  IF RND(1)<.1 THEN Y1=A(RND(4))
670  X1=X1+Y1:PROCXY(X1):IF ?FNS(XC,YC,6)=24
     THEN PRINT TAB(XC,YC);CHR$(224);:GOTO 560
680  X1=X1-Y1
690  R=RND(4):PROCXY(X1+A(R)):
     IF ?FNS(XC,YC,6)=24 THEN Y1=A(R):GOTO 670
700  PROCXY(X1+1):IF ?FNS(XC,YC,6)=24
     THEN Y1=1 :GOTO 670
710  PROCXY(X1-1):IF ?FNS(XC,YC,6)=24
     THEN Y1=-1:GOTO 670
720  PROCXY(X1+40):IF ?FNS(XC,YC,6)=24
     THEN Y1=40:GOTO 670
730  PROCXY(X1-40):IF ?FNS(XC,YC,6)=24
     THEN Y1=-40:GOTO 670
740  PRINT TAB(0,26);"YOU WIN !! ";:GOTO 760
750  PRINT TAB(0,26);"YOU LOSE !! ";
760  INPUT "ANOTHER GAME  ",Q$
770  IF LEFT$(Q$,1)="Y" THEN S=S-50:GOTO 380
780  MODE 7:*FX 4,0
790  GOTO 840
800  Y=40:RETURN
810  Y=-1:RETURN
820  Y=1:RETURN
830  Y=-40:RETURN
840  PRINT:END
850  DEFPROCXY(M)
860  YC=INT((M-32768)/40)
870  XC=M-32768-40*YC
880  ENDPROC
890  DEF FNS(X,Y,N)=HIMEM+(Y*40+X)*8+N
```

Mike James

PATTERNS

A chance to show off the BBC Micro's colour and Hi-Res graphics.



Lots of the programs in this collection use colour but, because other requirements of the programs dictate MODE 4, they are restricted to two-colour combinations. This program uses MODE 5 to give a splendid four-colour display on either Model A or Model B.

If you are using a black and white monitor with your computer don't feel left out and turn the page. The effects in monotone are still very pleasing as the different colours come out as various shades of grey.

The program listed is written so that it will run immediately on either model although this does initially limit the display to four colours. The colours selected are black, white, red and yellow for a crisp and striking effect (if you want to display a wider range of colours on a Model B machine see the notes below).

Random Symmetry

The program offers a choice of five displays. The first two are kaleidoscopes — reminiscent of the optical toys where you peer down a tube to see a continuously changing pattern of coloured shapes. The principles behind both the traditional kaleidioscope and this program are randomness and symmetry; the pattern changes at random but whatever happens in one quadrant is mirrored in the other three.

We can initiate two basic variants on the kaleidoscope theme. In the first, the pattern changes entirely at random and in the second, there are additional constraints introduced so that the pattern moves out from its centre.

In the next option, circles of varying diameter appear on the screen. Each one is made up of triangular segments of random colour. They have random mid-points and they can overlay

▶

each other filling the screen with bold, bright colour.

The final two programs make concentric ellipses resembling spider's webs appear on the screen. In the first version they appear in random colours at random positions and in the second, coloured webs spiral out from the centre and white webs spiral in from the outside. The full effect of the program is only apparent once the two sets of pattern have mingled — as they interwine, textures will appear in the pattern. When you break into this display it will finish a cycle before control is returned to you.

TECHNICAL DETAILS

As this program was written especially for the BBC Micro, it is different in structure from the conversions. In particular, procedures have been used as much as possible to make the program simply understood and easy to change. For example, the Kaleidoscope I and II programs both make use of a fundamental routine PROCSYM which will plot a point x,y in all four quadrants. You could produce another kaleidoscope program simply by altering the way in which PROCSYM is called with x,y and colour values. Notice that both kaleidoscope routines make use of block graphics in the form of CHR$ (224), a solid block defined in PROCINIT. This may seem a strange thing to do because MODE 5 is high resolution and using block graphics reduces the resolution to 40 by 32. However, if you try to produce a kaleidoscope program with points of the size MODE 5 uses, you will soon find that it takes a long time to fill the screen and changes will be almost invisible.

Another odd feature of the kaleidoscope procedures is the use of the VDU 5 command. This joins the text and graphics screens together allowing you to print a block anywhere on the screen. However, this feature isn't used by the program so you may be a little puzzled why we bother with it. The reaon is two-fold: first the VDU 5 command causes the speed of printing to be reduced and this makes the kaleidoscope display more effective and second, the VDU 5 commands gets round an apparent bug in BBC BASIC. If you print a character at location 19,31, (ie the bottom right-hand corner), then the text screen automatically scrolls even if you supress the carriage return/line feed. By using VDU 5 we can use the entire screen area *and* suppress scrolling.

PROGRAM STRUCTURE

Statement	Function	Action
Lines 10-20	Set Up	Select MODE 7 (for text display) and jump to initialisation routine.
Lines 30-130	Menu	Prints title, displays menu and asks user to make choice. Jumps to appropriate subroutine.
Lines 140-190	Kaleidoscopes	Selects MODE 5, joins together text and graphics area of screen. Notes which of two routines requested and jumps as appropriate.
Lines 200-270	Circles	MODE 5. Draws circles with random centres and random diameters in random colurs.
Lines 280-340	Web I	MODE 5. Draws concentric ellipses in random colours.
Lines 350-440	Web II	MODE 5. Draws ellipses stepping in from edge in white and ones in random colours stepping out from centre.
Lines 450-470	Character	Defines character 224 as solid square.
Lines 480-540	Symmetric Print	Allows character 224 to be echoed symmetrically in the four quadrants simultaneously.
Lines 550-590	Kaleidoscope I	Random kaleidoscope.
Lines 600-670	Kaleidoscope II	Kaleidoscope which steps outward from centre.
Lines 680-780	Plot circle	Routine to draw circles on screen. Line 750 contains equation.
Lines 790-870	Plot ellipse	Routine to draw ellipses on screen. Line 840 contains equation.

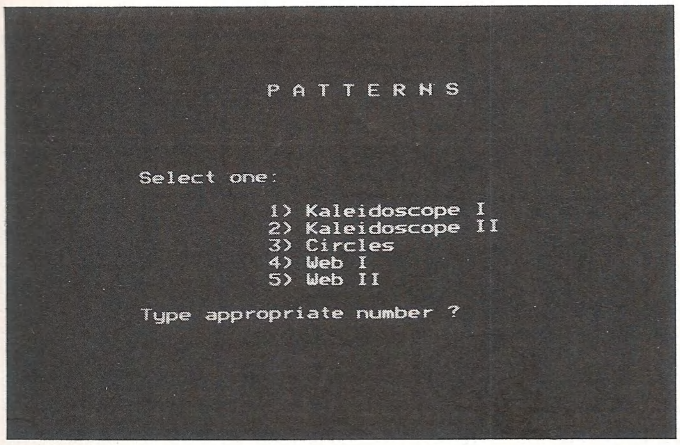
The rest of Patterns is fairly straightforward. The circles routine PROCCIR uses the PLOT 85 command to produce colour-filled triangular segments. The ellipse routine PROCELIPS uses a standard method for producing an ellipse by joining points on the ellipse by straight lines.

HOW TO USE THE PROGRAM

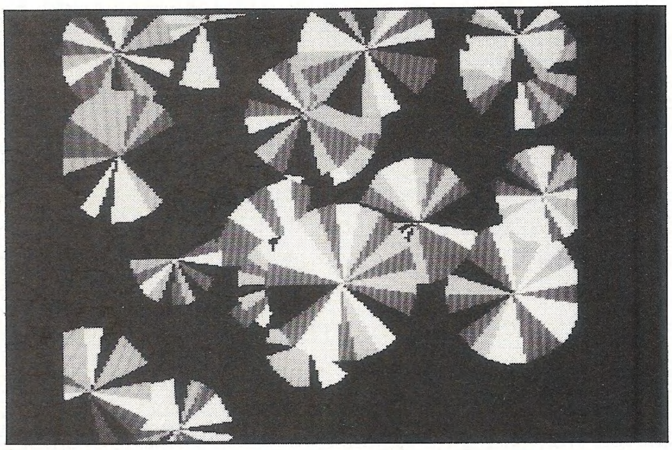
This one is virtually effortless. Select the number of your choice, type it in (not forgetting Return) and sit back. The display will continue until you stop it — by pressing *any* key. In the case of Web II, the program will not halt immediately but will finish its cycle.

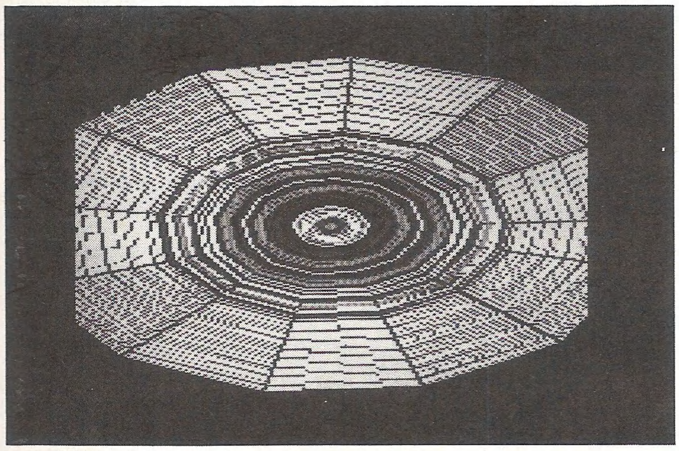
SCOPE FOR EXPERIMENTATION

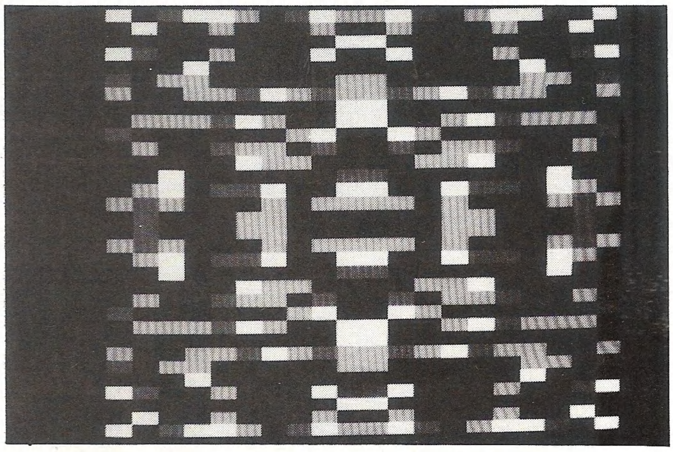
The program includes four colours — red, yellow and white as foreground colours and black as the background. However, you may prefer to program other colour combinations. If you are using a model B machine you can have eight by using MODE 2.



```
          P A T T E R N S


Select one:

          1) Kaleidoscope I
          2) Kaleidoscope II
          3) Circles
          4) Web I
          5) Web II

Type appropriate number ?
```







```
 10  MODE 7
 20  PROCINIT
 30  PRINT:PRINT TAB(10,5);"P A T T E R N S"
 40  PRINT TAB(0,10);"Select one:"
 50  PRINT
 60  PRINT TAB(10); "1) Kaleidoscope I"
 70  PRINT TAB(10);"2) Kaleidoscope II"
 80  PRINT TAB(10);"3) Circles"
 90  PRINT TAB(10);"4) Web I"
100  PRINT TAB(10);"5) Web II"
110  PRINT
120  INPUT "Type appropriate number ",A
130  ON A GOSUB 140,140,200,280,350
140  MODE 5
150  VDU 5
160  IF A=1 THEN PROCK1 ELSE PROCK2
170  B=INKEY(0)
180  IF B=-1 THEN GOTO 160
190  RUN
200  MODE 5
210  X=RND(1280)-1
220  Y=RND(1024)-1
230  R=RND(100)+100
240  C=RND(4)-1
250  PROCCIR(X,Y,R)
260  IF INKEY(0)<>-1 THEN RUN
270  GOTO 210
280  MODE 5
290  C=RND(4)-1
300  GCOL 0,C
310  R=RND(800)
320  PROCELIPS(R)
330  IF INKEY(0)<>-1 THEN RUN
340  GOTO 290
350  MODE 5
360  FOR R=1 TO 800 STEP 10
370  C=RND(4)-1
380  GCOL 0,C
390  PROCELIPS(R)
400  GCOL 4,0
410  PROCELIPS(800-R)
420  NEXT R
430  IF INKEY(0)<>-1 THEN RUN
440  GOTO 360
450  DEFPROCINIT
460  VDU 23,224,&FF,&FF;&FF,&FF,&FF,&FF,&FF,&FF
470  ENDPROC
480  DEFPROCSYM(X,Y,C)
490  GCOL 0,C
500  PRINT TAB(X,Y);CHR$(224);
510  PRINT TAB(19-X,Y);CHR$(224);
520  PRINT TAB(X,31-Y);CHR$(224);
530  PRINT TAB(19-X,31-Y);CHR$(224);
540  ENDPROC
550  DEFPROCK1
560  X=RND(10)-1
570  Y=RND(16)-1
580  C=RND(4)-1
590  PROCSYM(X,Y,C)
600  ENDPROC
610  DEFPROCK2
620  FOR X=10 TO 0 STEP -1
630  Y=RND(16)-1
640  C=RND(4)-1
650  PROCSYM(X,Y,C)
660  NEXT X
670  ENDPROC
680  DEFPROCCIR(X,Y,R)
690  INC=.3
700  MOVE X,Y
710  FOR T=0 TO PI*2 STEP INC
720  C=RND(4)-1
730  GCOL 0,C
740  MOVE R*COS(T)+X,R*SIN(T)+Y
750  PLOT 85, R*COS(T+INC)+X,R*SIN(T+INC)+Y
760  MOVE X,Y
770  NEXT T
780  ENDPROC
790  DEFPROCELIPS(R)
800  INC=.5
810  EX=.5
820  MOVE 640+R,512
830  FOR T=0 TO 2*PI STEP INC
840  PLOT 5,R*COS(T)+640,R*EX*SIN(T)+512
850  NEXT T
860  PLOT 5,640+R,512
870  ENDPROC
```

CRYSTAL BALL

There's no longer any need to cross the gypsy's palm with silver to have your fortune told — your BBC Micro will do it at the press of a key!

Crystal Ball is a light-hearted character assessment game which will amuse all your family and friends. You type in a name — your own name, the name of a friend or relative, even the name of a famous personality — and the computer will tell that person's fortune.

Name Or Number?

It's based on the idea of allocating numbers to each letter of the alphabet — A=1, B=2 to Z=26 — then adding them together repeatedly until only a single digit, in the range one to nine, remains. When a person's name is typed in, the computer works out their 'magic number', for example, 'JANE' = 10+9+14+5 = 30, 3+0 = 3, thus Jane's magic number will be 3.

HOW TO PLAY

The computer will ask you to type the name of the person whose character you want to assess and all you have to do then is to type it in.

At the end of each reading the computer will ask if you want more studies.

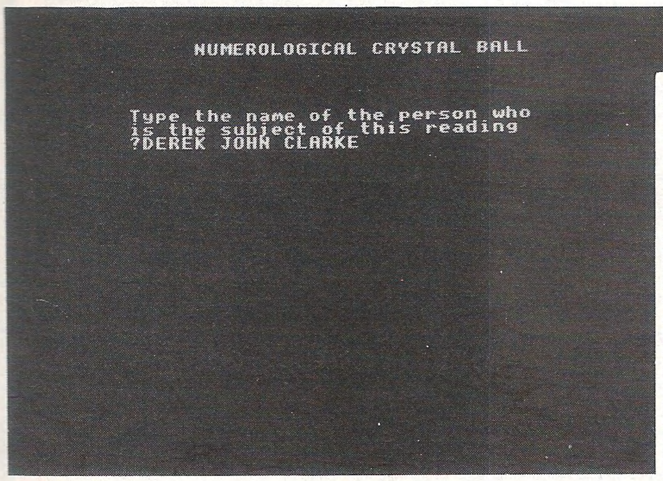
PROGRAM STRUCTURE

Statement	Function	Action
Lines 10-30	Set Up	Selects MODE 4 and clears screen.
Lines 40-90	Get Name	Prints title and asks for subject's name.
Lines 100-230	Arithmetic	Converts name to a single digit number by assigning value to each letter and adding repeatedly.
Lines 240-270	Prepare Screen	Displays subject's number and calls appropriate subroutine depending on its value (for 1 it goes to lines 290-350, for 2 it goes to lines 360-420 and so on).
Lines 280-970	Subroutines	Nine character readings, one of which is selected for each of the numbers 1 to 9.
Lines 980-1050	Repeat or Finish	Asks whether more readings are required; if YES, the program will restart, if NO, the program ends.

```
          NUMEROLOGICAL CRYSTAL BALL
Type the name of the person who
is the subject of this reading
?JANE
The subject's number is 3

Their main characteristics are:-

Sparkling character, full of
life.  Will draw good fortune
whatever they attempt.  Acute
intellect and wit.  Speaks with
charm and grace.  A natural
winner

Do you wish to do any more
studies?  Type Y/N
```

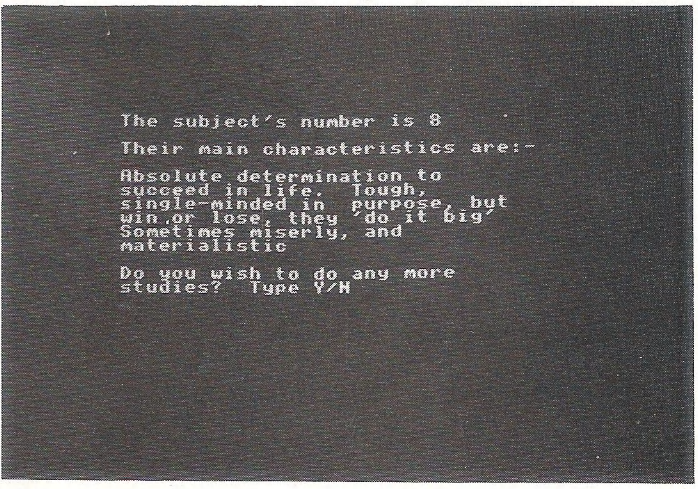
```
          NUMEROLOGICAL CRYSTAL BALL
Type the name of the person who
is the subject of this reading
?WINSTON CHURCHILL
The subject's number is 1

Their main characteristics are:-

At best enormously forceful
at worst annoyingly obstinate
Individualistic, having one
track minds. Massive ego
Either remarkable or a crashing
bore

Do you wish to do any more
studies?  Type Y/N
```

```
          NUMEROLOGICAL CRYSTAL BALL
Type the name of the person who
is the subject of this reading
?MAGGIE THATCHER
The subject's number is 8

Their main characteristics are:-

Absolute determination to
succeed in life.  Tough,
single-minded in  purpose, but
win or lose, they 'do it big'
Sometimes miserly, and
materialistic

Do you wish to do any more
studies?  Type Y/N
```



```
NUMEROLOGICAL CRYSTAL BALL


Type the name of the person who
is the subject of this reading
?DEREK JOHN CLARKE
```

Left: The question, who's the subject?



```
The subject's number is 8

Their main characteristics are:-

Absolute determination to
succeed in life.  Tough,
single-minded in  purpose, but
win or lose, they 'do it big'
Sometimes miserly, and
materialistic

Do you wish to do any more
studies?  Type Y/N
```

Right: The answers are sometimes uncomfortably accurate!

```
  10  REM**PROGRAM FOR NUMEROLOGICAL
      CHARACTER ASSESSMENT
  20  MODE 4
  30  CLS
  40  PRINT
  50  PRINT"     NUMEROLOGICAL CRYSTAL BALL"
  60  PRINT TAB(0,6);"Type the name of
      the person who"
  70  PRINT"is the subject of this reading"
  80  S=0:B=0
  90  INPUT A$
 100  FOR A=1 TO LEN(A$)
 110  IF MID$(A$,A,1)=" "THEN S=S+1
 120  B=B+ASC(MID$(A$,A,1))
 130  NEXT A
 140  Z=B-S*32-((LEN(A$)-S)*64)
 150  B=INT(Z/100)
 160  C=INT((Z-B*100)/10)
 170  D=INT(Z-B*100-C*10)
 180  E=B+C+D
 190  IF E<=9 THEN 240
 200  F=INT(E/10)
 210  G=E-F*10
 220  E=F+G
 230  GOTO 190
 240  CLS
 250  PRINT TAB(0,6);"The subject's
      number is ";E
 260  PRINT:PRINT"Their main
       characteristics are:-":PRINT
 270  ON E GOSUB 290,360,440,510,590,660,
      750,830,900
 280  GOTO 980
 290  PRINT"At best enormously forceful"
 300  PRINT"at worst annoyingly obstinate"
 310  PRINT"Individualistic, having one"
 320  PRINT"track minds. Massive ego"
 330  PRINT"Either remarkable or a crashing"
 340  PRINT"bore."
 350  RETURN
 360  PRINT "Exhibits feminine
      characteristics"
 370  PRINT"Reticent, willing to compromise"
 380  PRINT"May seem placid and"
 390  PRINT"ingratiating, but is
      plotting."
 400  PRINT"all the time.  Can be"
 410  PRINT"genuinely sweet and has an"
 420  PRINT"appealing quiet side."
 430  RETURN
 440  PRINT"Sparkling character, full of"
 450  PRINT"life.  Will draw good fortune"
 460  PRINT"whatever they attempt.  Acute"
 470  PRINT"intellect and wit.  Speaks with"
 480  PRINT"charm and grace.  A natural"
 490  PRINT"winner."
 500  RETURN
 510  PRINT"Inauspicious, dull and lacking"
 520  PRINT"in imagination.  Self-righteous"
 530  PRINT"and respectable to a fault"
 540  PRINT"At best can be a pillar of"
 550  PRINT"society, capable of"
 560  PRINT"making a positive contribution"
 570  PRINT"to it."
 580  RETURN
 590  PRINT"Make excellent (if not"
 600  PRINT"faithful) bedmates. Extremely"
 610  PRINT"attractive and energetic."
 620  PRINT"Multiple interests and talents"
 630  PRINT"Dangerous to know, but fun"
 640  PRINT"to watch!"
 650  RETURN
 660  PRINT"A paragon of domesticity and"
 670  PRINT"harmony with an even temper"
 680  PRINT"Fair minded and slow to anger"
 690  PRINT"Make excellent wife/husband"
 700  PRINT"conscientious parent and loyal"
 710  PRINT"friend.  Fun to be with (if you"
 720  PRINT"can put up with their"
 730  PRINT"occasional tendency to gossip)."
 740  RETURN
 750  PRINT"Secretive, stand-offish"
 760  PRINT"disciplined and haughty"
 770  PRINT"Mysterious, often hiding"
 780  PRINT"something - unhappiness"
 790  PRINT"world-weariness or disillusionment"
 800  PRINT"Can be bitter, disdainful"
 810  PRINT"and sarcastic."
 820  RETURN
 830  PRINT"Absolute determination to"
 840  PRINT"succeed in life.  Tough,"
 850  PRINT"single-minded in  purpose, but"
 860  PRINT"win or lose, they 'do it big'"
 870  PRINT"Sometimes miserly, and"
 880  PRINT"materialistic"
 890  RETURN
 900  PRINT"A wise counsellor, spiritual"
 910  PRINT"leader, seeker-after and"
 920  PRINT"teacher-of truth.  Idealist"
 930  PRINT"in the highest sense of the"
 940  PRINT"word.  Strongly passionate"
 950  PRINT"and possessed of durable will"
 960  PRINT"A remarkable person"
 970  RETURN
 980  PRINT:PRINT"Do you wish to do any more"
 990  PRINT"studies?  Type Y/N"
1000  Z$=GET$
1010  IF Z$="Y" THEN 10
1020  IF Z$="N" THEN STOP
1030  PRINT"RESPONSE NOT RECOGNISED"
1040  GOTO 980
1050  END
```

Subscriptions

Are you looking for a more personal approach to computing?

You are . . . then Computing Today is the magazine for you! Packed full of feature articles, projects, general topics, news and reviews, Computing Toady is aimed at readers who want to get more out of their microcomputer.

The latest ABC circulation figures show Computing Today has increased its readership by 85% over the previous year — great news for us at CT. However, the ever increasing demand for Computing Today has meant that, despite our printing more each month, some readers seem to be missing out on their regular copy.

If you would like to ensure a regular supply for the next twelve months, each issue lovingly wrapped and posted to you, nothing could be simpler. Just fill in the form below, cut it out and send it with your cheque or Postal Order (made payable to ASP Ltd) to:

Computing Today Subscriptions,
513 London Road,
Thornton Heath,
Surrey CR4 6AR

Alternatively you can pay by Access or Barclaycard in which case simply fill in your card number, sign the form and send it off. Do NOT send your card.

Do yourself a favour, make 1982 the year you start to take Computing Today every month, we'll give you a truly

Personal Approach To Microcomputing.

SUBSCRIPTION ORDER FORM

Cut out and SEND TO :
Computing Today Subscriptions
513, LONDON ROAD,
THORNTON HEATH,
SURREY,
ENGLAND.

Please commence my personal subscription to Computing Today with the .... issue.

SUBSCRIPTION RATES
(tick ☐ as appropriate)

£11.50 for 12 issues U.K ☐
£15.15 for 12 issues overseas surface ☐
£34.75 for 12 issues Overseas Air Mail ☐

I am enclosing my (delete as necessary) Cheque/Postal Order/International Money Order for £..........
(made payable to ASP Ltd)
OR
*Debit my Access/Barclaycard**
*(*delete as necessary)*

Access
We welcome Access

BARCLAYCARD
VISA

Please use BLOCK CAPITALS and include post codes.

Name (Mr/Mrs/Miss) ....
delete accordingly

Address ....

....

....

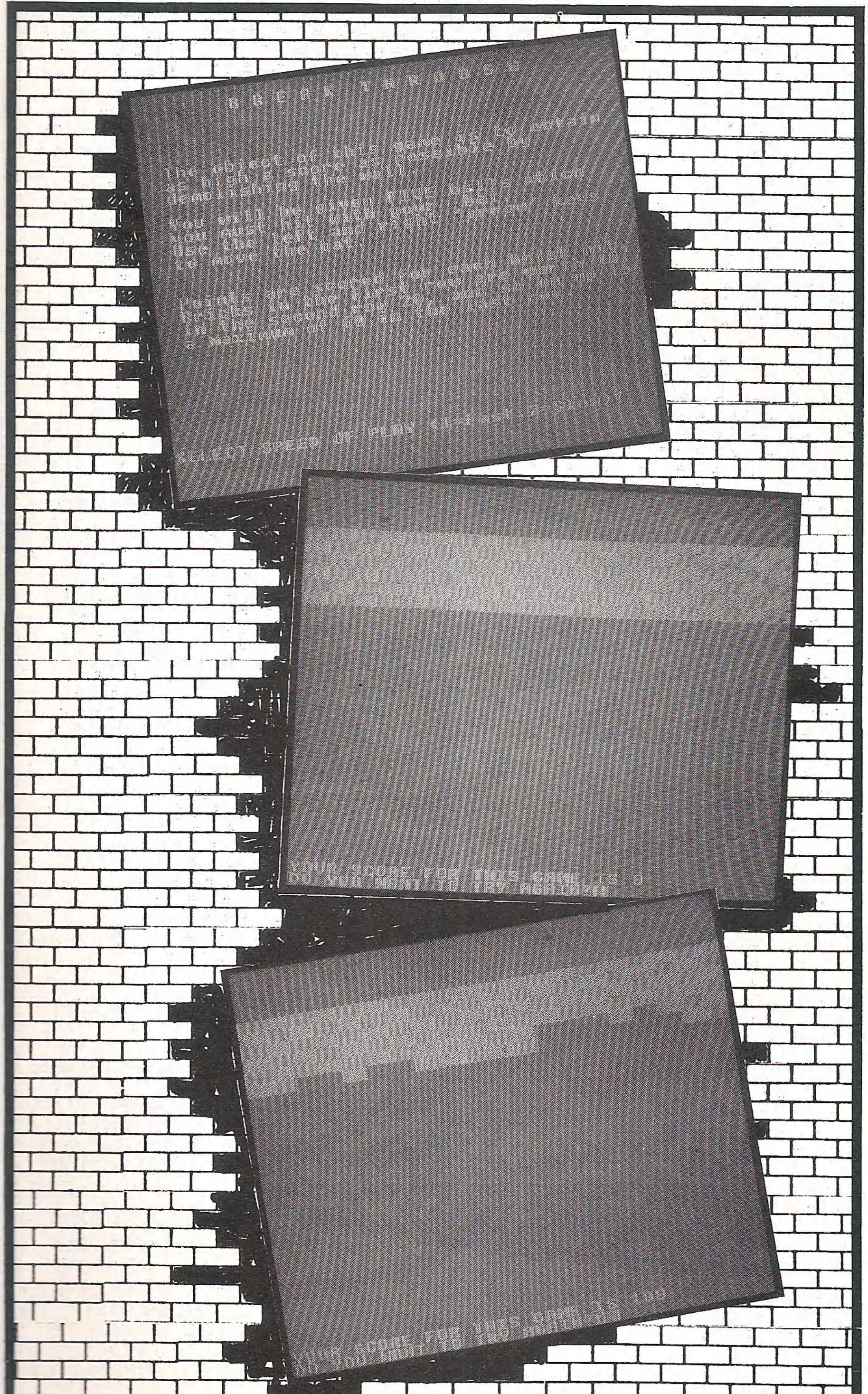
Signature ....

Date ....

Mike James

BREAK THROUGH

As soon as you've demolished one wall with this game, you can always build another one.



Here's a program to test your reaction times and see just how good your judgement is. It's an old favourite — for which I make no apologies since there's good reason for its popularity. It will keep the kids happily occupied for hours, but don't fall into the trap of 'just having a quick go at it' — its compulsive and you'll be stuck there for hours!

Another Brick In The Wall

There are six rows of bricks and you have a bat and five balls. You have to break through the barrier, aiming to score as many points as possible. The front line of bricks are worth ten points each, and each succeeding row is worth ten more — so for every one you wipe out in the back row you'll add 60 to your tally. A really good move is to trap the ball between the back row and the edge of the screen and see it ricochet back and forth giving you a massive score.

In this version you can set the speed of play — fast or slow. The display is in colour (yellow on blue) and uses sound — a satisfying noise when the ball hits your bat or the wall but a distressing sound accompanying your ball as it goes out of play. The score is displayed continuously and is noted at the end of each game.

TECHNICAL DETAILS

This program has been written from scratch for the BBC Micro so it has a tighter structure than some of the conversions and reflects the way BBC BASIC can be used to advantage.

The program is nothing more than a list of PROC calls and as such should be easy to understand and change. The

▶

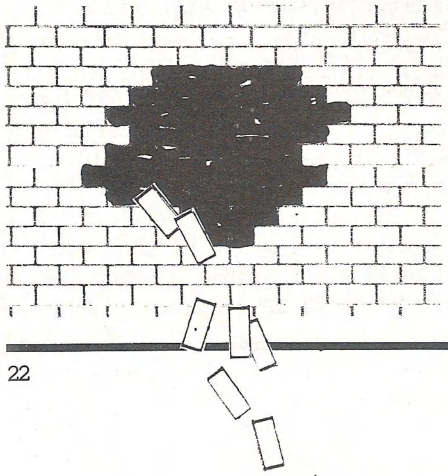
techniques used in each PROC are either straightforward or are similar to ones used in the converted programs. The only new feature is the changing of the repeat delay and rate to give improved control over the bat. If you take out lines 240 and 250 you can see the problem of playing a moving graphics games controlled by a standard keyboard. When you press one of the 'arrow' keys the bat moves once followed by a delay, and then the repeat facility causes it to move continuously at the repeat speed. This makes the bat very difficult to control.

What we would really like to happen when a key is pressed is for the repeat to occur at once and very fast. This can be achieved by using *FX 11,1 and *FX 12,1 which changes the repeat delay and the repeat period to 1 millisecond. The command *FX 12,0 will restore the keyboard to normal function.

HOW TO PLAY

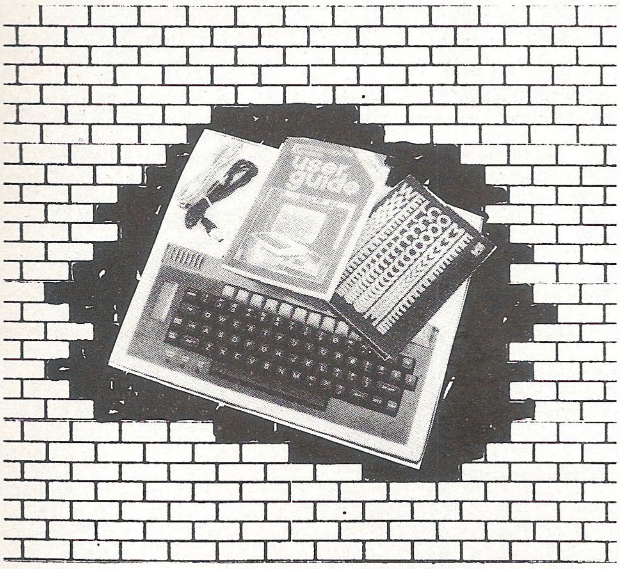
Select the speed at which you want to play (1 is faster than 2). The bat can move right and left through the use of the appropriate cursor control arrows (at the top right of the keyboard). The repeat mechanism is used for this game so if you keep your finger on an arrow it will continue to move; the speed of the repeats is very fast so you need to be very nimble fingered.

Do not try to exit from the program by breaking in. At the end of each game the computer asks you if you wish to play again. The key repeat mechanism is restored to normal by the program *if* you exit properly — by replying 'NO' to this question.



PROGRAM STRUCTURE

Statement	Function	Action
Lines 10-20	Set Up	Selects MODE 4 and sets colour (red foreground and blue background).
Lines 30-140	Program Core	Calls the different procedures defined later in the program — that at line 30 prints the opening text, that at line 40 sets up the constants, that at line 60 draws the screen, that at line 70 prints messages during the game and that at line 80 controls play. Initial values and flags are set and the bottom two lines of the screen are blanked. Line 110 generates a dull sound whenever the ball goes out of play.
Lines 150-160	Restore Keyboard	Enables keyboard to be restored to normal operation at end of play.
Lines 170-210	End Game	Prints score message at end of game and asks if further play is required.
Lines 220-250	Set Keyboard	Enables use of arrow keys and sets high repeat rate on keyboard.
Lines 260-370	DEFPROC-CODE	Defines characters for graphics display and sets up constants.
Lines 380-590	DEFPROC-TEXT	Displays text at beginning of game.
Lines 600-660	DEFPROC-SCR	Prints graphics display on screen.
Lines 670-830	DEFPROC-PLAY	Controls play when arrow keys are pressed.
Lines 840-860	DEFPROC-BAT	Displays and moves bat.
Lines 870-970	DEFPROC-BALL	Displays 'ball'; logic to move and bounce ball.
Lines 980-1000	DEFPROC-BEEP	Sounds 'beep' noise when ball bounces.
Lines 1010-1100	DEFPROC-DB	Checks to discover whether the 'ball' has hit the wall and removes a double character block when a hit occurs. Keeps score.
Lines 1110-1140	DEFPROC-PLAYMES	Prints play messages for score and indicates which ball is in play.
Lines 1150-1240	ODDROW & EVENROW	Displays rows of bricks with alternating patterns.
Line 1250	Address	Defines x and y co-ordinates for screen address.



```
  10  MODE 4
  20  VDU 19,0,4,0,0,0:VDU 19,1,3,0,0,0
  30  PROCTEXT
  40  PROCCODE
  50  B=5:SC=0
  60  PROCSCR
  70  PROCPLAYMES
  80  PROCPLAY
  90  B=B-1
 100  E=0:F=0
 110  SOUND 2,-15,50,15
 120  PRINT TAB(0,30);SPC(40);
 130  PRINT TAB(0,28);SPC(40);
 140  IF B<>0 THEN GOTO 70
 150  *FX 4 0
 160  *FX 12 0
 170  PRINT TAB(0,31);SPC(39);
 180  PRINT TAB(0,30);"YOUR SCORE FOR
      THIS GAME IS ";SC
 190  INPUT"DO YOU WANT TO TRY AGAIN",A$
 200  A$=LEFT$(A$,1)
 210  IF A$="N" THEN MODE 7:END
 220  *FX 11 1
 230  *FX 12 1
 240  *FX 4 1
 250  GOTO 50
 260  DEFPROCCODE
 270  VDU 23,224,&FF,&FF,&FF,&FF,&FF,&FF,
      &FF,&FF
 280  VDU 23,225,&CC,&33,&CC,&33,&CC,&33,
      &CC,&33
 290  L=3
 300  *FX 4 1
 310  D=0
 320  E=0
 330  F=0
 340  SC=0
 350  *FX 11 1
 360  *FX 12 1
 370  ENDPROC
 380  DEFPROCTEXT
 390  CLS
 400  PRINT
 410  PRINT TAB(8);"B R E A K  T H R O U G H"
 420  PRINT:PRINT:PRINT
 430  PRINT
 440  PRINT "  The object of this game
      is to obtain"
 450  PRINT "  as high a score as possible by"
 460  PRINT "  demolishing the wall."
 470  PRINT
 480  PRINT "  You will be given FIVE
      balls which"
 490  PRINT "  you must hit with your  bat."
 500  PRINT "  Use the left and right
      'arrow' keys"
 510  PRINT "  to move the bat."
 520  PRINT:PRINT
 530  PRINT "  Points are scored for
      each brick hit."
```

```
 540  PRINT "  Bricks in the first row
      are worth 10,"
 550  PRINT "  in the second row 20, and
      so on up to"
 560  PRINT "  a maximum of 60 in the
      last row."
 570  PRINT TAB(0,28);"SELECT SPEED OF
      PLAY (1=Fast,2=Slow)";:INPUT H
 580  IF H<>1 AND H<>2 THEN GOTO 570
 590  ENDPROC
 600  DEFPROCSCR
 610  CLS
 620  FOR Y=4 TO 8  STEP 2
 630  PROCODDROW(Y)
 640  PROCEVENROW(Y+1)
 650  NEXT Y
 660  ENDPROC
 670  DEFPROCPLAY
 680  X=19:Y=28
 690  PROCBAT(X,Y)
 700  FOR I=0 TO 1000+RND(500):NEXT I
 710  BY=10:BX=5+RND(5)
 720  HV=(RND(2)-1)*2-1:VV=1
 730  PROCBAT(X,Y)
 740  D=D+1
 750  IF D=H THEN D=0:PROCBALL
 760  IF E=1 THEN ENDPROC
 770  A=INKEY(0):IF A=-1 THEN GOTO 730
 780  *FX 15,0
 790  A=2*(A-&88)-1
 800  IF A<>-1 AND A<>1 THEN 730
 810  IF X+A<0 OR X+A>38-L THEN GOTO 730
 820  X=X+A
 830  GOTO 730
 840  DEFPROCBAT(X,Y)
 850  PRINT TAB(X,Y);" ";
      STRING$(L,CHR$(244));" ";
 860  ENDPROC
 870  DEFPROCBALL
 880  E=0
 890  PRINT TAB(BX,BY);" ";
 900  BX=BX+HV:BY=BY+VV
 910  IF BX<1  OR BX>38  THEN HV=-HV:
      BX=BX+HV:PROCBEEP:F=0
 920  IF BY<0 THEN VV=-VV:BY=BY+VV:PROCBEEP:F=0
 930  IF ?FNS(BX,BY,0)<>0  THEN PROCBEEP:
      PROCDB(BX,BY)
 940  IF BY=30 THEN E=1
 950  PRINT TAB(BX,BY);CHR$(224);
 960  PRINT TAB(15,31);SC;
 970  ENDPROC
 980  DEFPROCBEEP
 990  SOUND 2,-15,150,1
1000  ENDPROC
1010  DEFPROCDB(BX,BY)
1020  VV=-VV
1030  IF BY>20 THEN F=0:ENDPROC
1040  S=-1
1050  IF INT(BX/2)*2=BX THEN S=1
1060  PRINT TAB(BX+S,BY);" ";
1070  SC=SC+(10-BY)*10
1080  IF F=0 THEN F=1:ENDPROC
1090  VV=-VV
1100  ENDPROC
1110  DEFPROCPLAYMES
1120  PRINT TAB(0,31);"YOUR  SCORE=";
1130  PRINT TAB(20,31);"BALL = ";6-B;
1140  ENDPROC
1150  DEFPROCODDROW(Y)
1160  FOR I=0 TO 39   STEP 4
1170  PRINT TAB(I,Y);CHR$(224);CHR$(224);
      CHR$(225);CHR$(225);
1180  NEXT I
1190  ENDPROC
1200  DEFPROCEVENROW(Y)
1210  FOR I=0 TO 39   STEP 4
1220  PRINT TAB(I,Y);CHR$(225);CHR$(225);
      CHR$(224);CHR$(224);
1230  NEXT I
1240  ENDPROC
1250  DEF FNS(X,Y,N)=HIMEM+(Y*40+X)*8+N
```

Converted from a program by A Lacy

WORD SQUARE

Find the hidden words in this the ultimate crossword that goes up, down, and all around. Confused . . . you will be!

PROGRAM STRUCTURE

Statement	Function	Action
Lines 10-50	Initialise	Defines arrays to be used.
Lines 60-180	Instructions	Prints title and instructions.
Lines 190-260	Shuffle	Shuffles co-ordinates.
Lines 270-410	Input	Requests input, checks for end of list marker and calls routine to test each word for length and content.
Lines 420-440	Sort	Prints message on screen and calls routine to sort words by length.
Lines 450-650	Fit Words	Puts words in random directions and tests to ensure that they fit. Meanwhile prints message.
Lines 660-720	Array	Assembles words into array.
Lines 730-770	Try Again	Tries again if failed to achieve a fit.
Lines 780-850	Failed	Failed to fit entire array. Gives choice of re-starting or seeing partial array.
Lines 860-870	Loop End	End of word fit loop.
Lines 880-970	First Print	Prints Wordsquare without random letters.
Lines 980-1160	Final Print	Requests title then prints Wordsquare (plus random letters) with list of words or dashes alongside.
Lines 1170-1230	Input Tests	If words too long or if characters other than letters input, message displayed.
Lines 1240-1330	Directional Logic	Logic to move letters in all possible directions.
Lines 1340-1400	Sort	Routine to sort words by length.

VARIABLES

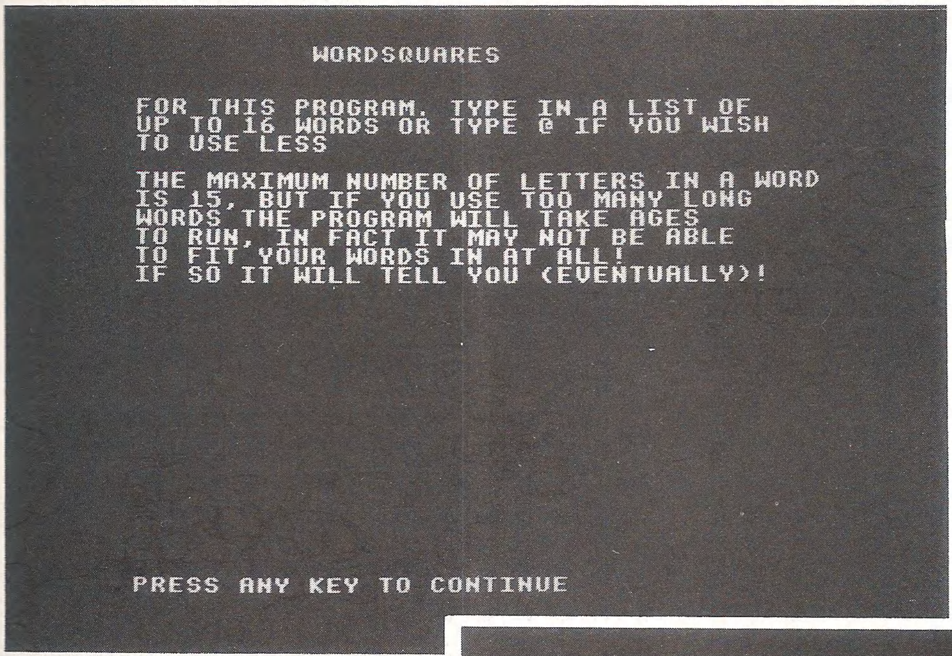
A,C,D,G,P,X,I	Temporary storage integers.
C(n)	List of shuffled vertical co-ordinates.
C1,C2	Temporary storage for vertical co-ordinates.
D(n)	List of shuffled directions.
Q$	Temporary storage for single letter replies.
R(n)	List of shuffled horizontal co-ordinates.
R1,R2	Temporary storage for horizontal co-ordinates.
S$	Temporary storage for inputted words.
S$(n,m)	The string array representing the Wordsquare.
T$	Title.
W$(n)	List of inputted words.
W1,W2	Pointers into W$(n).
X1,X2	Used in row and column shuffle.

The idea behind this game is that you give the computer a list of words on a particular theme and it hides them in a 15 by 15 grid of letters. First, the computer positions the words it is given, writing them in one of eight directions — forwards (that is, from left to right), backwards (from right to left), downwards (going vertically down the grid),upwards (going vertically up the grid) or along a diagonal in one of the four possible directions. The computer then fills in all the remaining spaces with random letters. The randomising method ensures that no two puzzles are similar even though they may contain identical words.

Now You See Them . . .

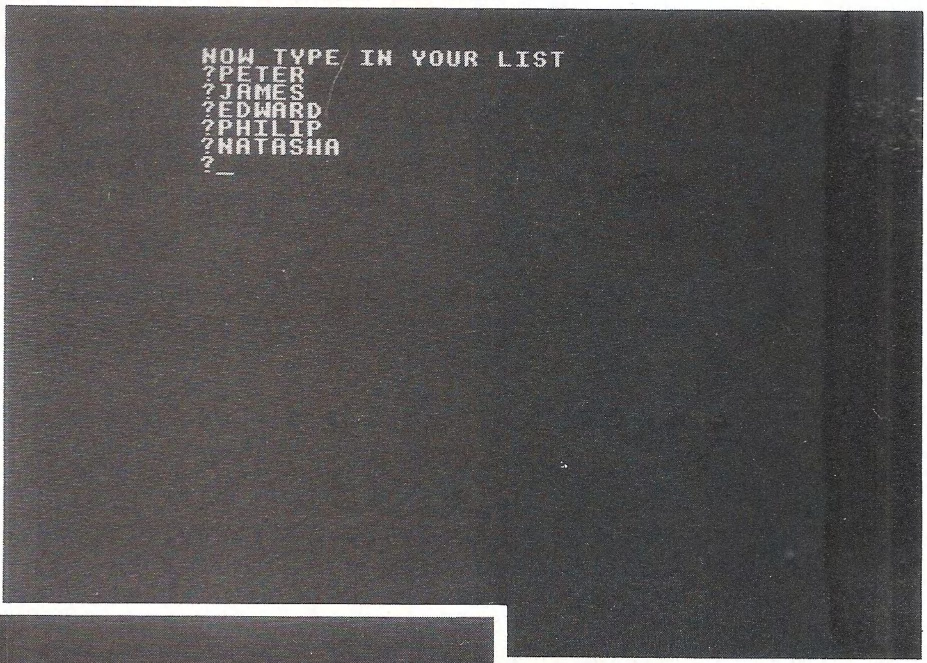
The Wordsquare can be displayed either with or without the list of words given originally alongside it. If you choose not to see the wordlist then a number of dashes corresponding to the actual words are printed so that you can tell how many letters there are in each word. However, if you are playing the game this way, the title you give the Wordsquare is very important as it indicates the sort of words to look for — after all, when all the random letters are added you may well be able to find words which were not actually intended!

You can select up to 16 words and each word can be up to 15 letters long. However, if you have a lot of long words the program will take longer to run and, if you stretch it too far, the computer may come back and tell you that it cannot cope.



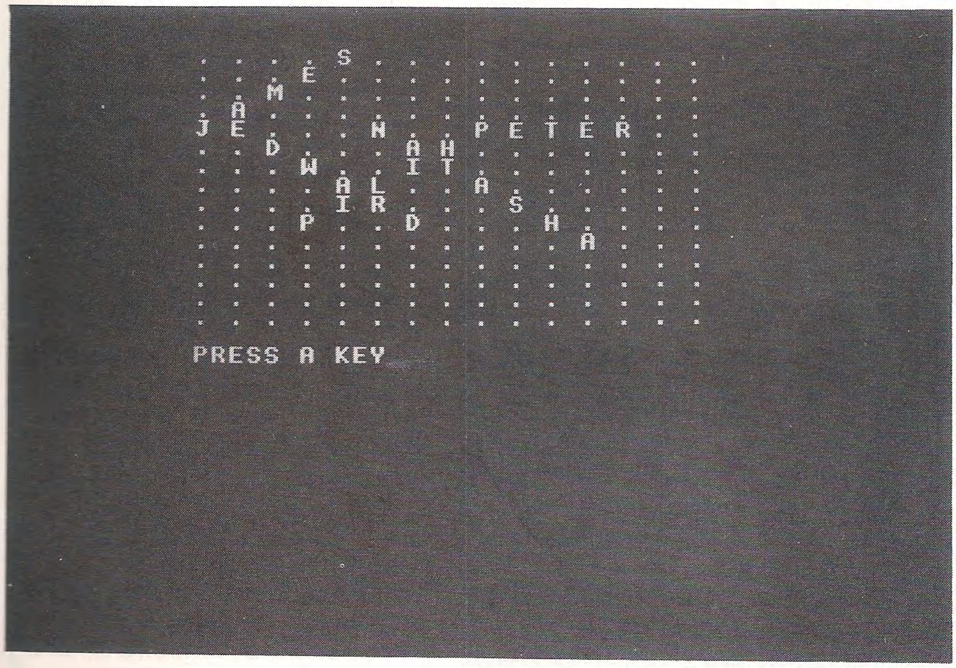
```
              WORDSQUARES

      FOR THIS PROGRAM, TYPE IN A LIST OF
      UP TO 16 WORDS OR TYPE @ IF YOU WISH
      TO USE LESS

      THE MAXIMUM NUMBER OF LETTERS IN A WORD
      IS 15, BUT IF YOU USE TOO MANY LONG
      WORDS THE PROGRAM WILL TAKE AGES
      TO RUN, IN FACT IT MAY NOT BE ABLE
      TO FIT YOUR WORDS IN AT ALL!
      IF SO IT WILL TELL YOU (EVENTUALLY)!







      PRESS ANY KEY TO CONTINUE
```

The introductory text and instructions displayed by the program on RUNning.



```
NOW TYPE IN YOUR LIST
?PETER
?JAMES
?EDWARD
?PHILIP
?NATASHA
?_
```

Entering the words which we wish to be hidden in the Wordsquare. Try to keep the words entered of the same sort of category (for example, people's christian names) as otherwise the puzzle may prove very hard to solve.



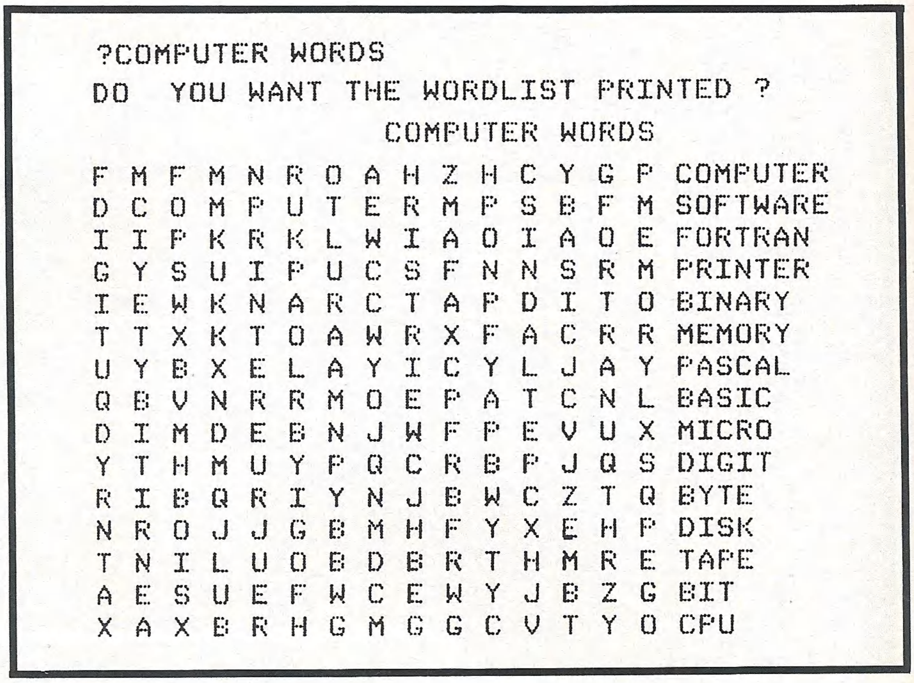
```
. . . . S . . . . . . . . . .
. . . E . . . . . . . . . . .
. . M . . . . . . . . . . . .
. A . . . . . . . . . . . . .
J E . . . N . . P E T E R . .
. . D . . . A H . . . . . . .
. . . W . . I T . . . . . . .
. . . . A L . . A . . . . . .
. . . . I R . . . S . . . . .
. . . P . . D . . . H . . . .
. . . . . . . . . . . A . . .
. . . . . . . . . . . . . . .
. . . . . . . . . . . . . . .
. . . . . . . . . . . . . . .
. . . . . . . . . . . . . . .

PRESS A KEY
```

The program now builds the square with our words embedded in all directions.

The final display, see over page, has the completed Wordsquare with its title and the optional list of embedded words printed out.

Even with a moderate list, the computer takes some time to prepare its square. While it is processing the problem it prints 'Thinking Hard', and repeats 'Hard' to let you know it has not forgotten the problem. If it can't manage your list of words it will let you know and will give you the option of accepting the partially completed puzzle. It could be that it gave up very near the end in which case the Wordsquare will already be a challenge to unravel.

The computer sorts the words by their length and produces its list with the longest ones at the top. This is because it places the longest words into the grid first and then the shorter ones are fitted around and through them. One letter can belong to more than one word — as in a crossword — except that it can be read along the diagonals as well.



```
?COMPUTER WORDS
DO  YOU WANT THE WORDLIST PRINTED ?
                        COMPUTER WORDS
F M F M N R O A H Z H C Y G P COMPUTER
D C O M P U T E R M P S B F M SOFTWARE
I I P K R K L W I A O I A O E FORTRAN
G Y S U I P U C S F N N S R M PRINTER
I E W K N A R C T A P D I T O BINARY
T T X K T O A W R X F A C R R MEMORY
U Y B X E L A Y I C Y L J A Y PASCAL
Q B V N R R M O E P A T C N L BASIC
D I M D E B N J W F P E V U X MICRO
Y T H M U Y P Q C R B P J Q S DIGIT
R I B Q R I Y N J B W C Z T Q BYTE
N R O J J G B M H F Y X E H P DISK
T N I L U O B D B R T H M R E TAPE
A E S U E F W C E W Y J B Z G BIT
X A X B R H G M G G C V T Y O CPU
```

Checking Or Cheating?

When the puzzle is constructed it will be displayed on the screen without the addition of the random letters. This is very useful for checking purposes — or for if you want to cheat; when others come to you after spending hours searching for the words you can pick up the puzzle and solve it in moments. If you really do not want this preview, you could delete lines 900-950, but instead of such a drastic measure you can always move on very rapidly so that you don't notice much.

This puzzle has the added advantage that, if you have a printer, you can easily make a 'hard copy' — just the job for train journeys or dentist's waiting rooms.

TECHNICAL DETAILS

This is a fairly straightforward program. The only real difficulty encountered converting it from TRS-80 BASIC was the way FOR loops were jumped out of and re-started. Some BASICs protect you from this most evil practice by checking to see if there are any active FOR loops using the same index variable every time you start a new FOR loop. If there are then the old FOR loop is cleared up and the new one takes its place.

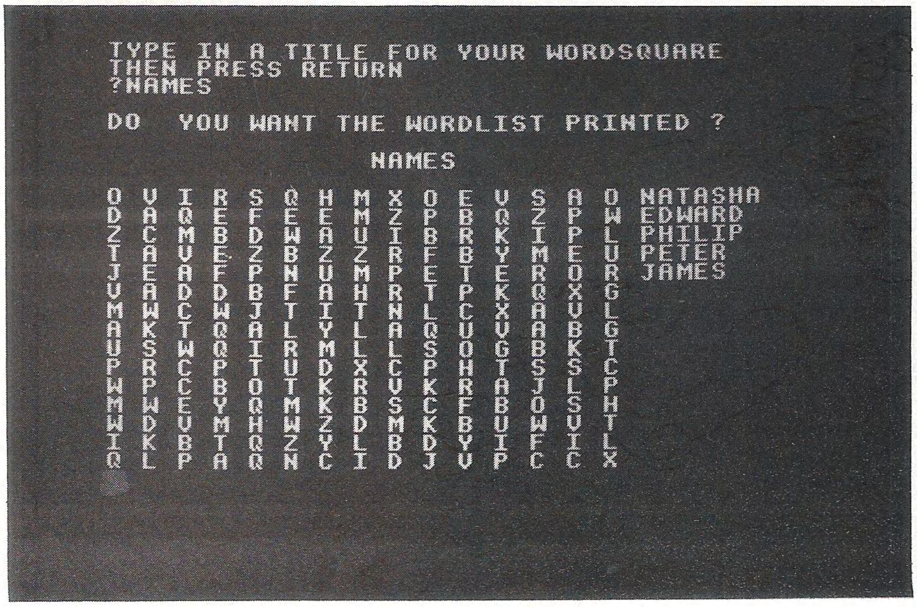
The BBC Micro, however, won't let you be this sloppy and requires each FOR loop to be completed before a new one, using the same index variable, can be started. The premature ending of FOR loops in the original version has been overcome either by the use of end of loop flags or by setting the index variable to its maximum value and jumping to the correct NEXT statement.

HOW TO PLAY

The computer will ask you to type in your chosen words. Type them one at a time and press Return after each. The computer will prompt for each with a '?'. If you want to finish the list at any point, type '**@**'.

After you've typed in the list, the computer will take a little time to 'hide' them prior to displaying the Wordsquare *before* the addition of the random letters that will obscure the initial words. The computer will then ask whether you wish to have a list of the words printed out. You will also be asked to give the puzzle a title — to indicate the type of words which are hidden in it.

If you have a printer interfaced to your BBC Micro, type 'CTRL B' to make a hard copy of the Wordsquare.



```
TYPE IN A TITLE FOR YOUR WORDSQUARE
THEN PRESS RETURN
?NAMES

DO  YOU WANT THE WORDLIST PRINTED ?

                NAMES

O V I R S Q H M X O E V S A O NATASHA
D A Q E F E E M Z P B Q Z P W EDWARD
Z C M B D W A U I B R K I P L PHILIP
T A V E Z B Z Z R F B Y M E U PETER
J E A F P N U M P E T E R O R JAMES
V A D D B F A H R T P K Q X G
M W C W J T I T N L C X A V L
A K T Q A L Y L A Q U V A B G
U S W Q I R M L L S O G B K T
P R C P T U D X C P H T S S C
W P C B O T K R U K R A J L P
M W E Y Q M K B S C F B O S H
W D V M H W Z D M K B U W V T
I K B T Q Z Y L B D Y I F I L
Q L P A Q N C I D J V P C C X
```

```
  10  REM WORDSQUARE
  20  REM
  30  REM INITIALISE
  40  CLS
  50  DIM C(15),R(15),D(8),W$(17),S$(15,15),
      R2(15),C2(15)
  60  PRINT: PRINT TAB(10);"WORDSQUARE"
  70  PRINT
  80  PRINT
  90  PRINT "FOR THIS PROGRAM, TYPE IN A
      LIST OF"
 100  PRINT "UP TO 16 WORDS OR TYPE @ IF
      YOU WISH"
 110  PRINT "TO USE LESS"
 120  PRINT:PRINT "THE MAXIMUM NUMBER OF
      LETTERS IN A WORD"
 130  PRINT "IS 15, BUT IF YOU USE TOO
      MANY LONG"
 140  PRINT "WORDS THE PROGRAM WILL TAKE AGES"
 150  PRINT "TO RUN, IN FACT IT MAY NOT BE"
 160  PRINT "ABLE TO FIT YOUR WORDS IN AT ALL"
 170  PRINT "IF SO IT WILL TELL YOU
      (EVENTUALLY)!"
 180  PRINT TAB(0,30);"PRESS ANY KEY TO
      CONTINUE":A$=GET$
 190  REM**SHUFFLE COORDINATES
 200  FOR A=1 TO 15:C(A)=A:R(A)=A:IF
      A<9 THEN D(A)=A
 210  NEXT
 220  FOR I=1 TO 2
 230  FOR A=1 TO 15
 240  R1=RND(15):R2=RND(15):X1=R(R1):
      X2=C(R2):R(R1)=R(A):C(R2)=C(A):R(A)=X1:
      C(A)=X2
 250  NEXT
 260  NEXT I
 270  REM**INPUT WORDLIST
 280  CLS:PRINT
 290  PRINT"NOW TYPE IN YOUR LIST"
 300  E=0
 310  FOR W=1 TO 16
 320  IF E=1 THEN GOTO 390
 330  INPUT S$:IF S$="@" THEN E=1:V=W-1 :S$=""
 340  W$(W)=S$
 350  REM**TEST THE WORD FOR LENGTH AND
      CONTENT,S IS ERROR FLAG
 360  S=0
 370  IF E=0 THEN GOSUB 1180
 380  IF S=1 THEN 330
 390  NEXT W
 400  IF E=0 THEN V=16
 410  W=V
 420  CLS:PRINT:PRINT "THIS COULD TAKE
      ME A FEW MINUTES"
 430  REM**SORT WORDS, LONGEST FIRST
 440  GOSUB 1350
 450  REM**CHOOSE THE NEXT WORD
 460  FOR W1=1 TO W
 470  REM**RANDOMISE DIRECTIONS
 480  FOR A=1 TO 8:R=RND(8):X=D(R):
      D(R)=D(A):D(A)=X:NEXT
 490  REM**CHOOSE A COORDINATE
 500  FOR R=1 TO 15
 510  PRINT "THINKING ";
 520  FOR C=1 TO 15
 530  REM**CHOOSE A DIRECTION
 540  FOR D=1 TO 8
 550  R2=R(R):C2=C(C)
 560  REM**CAN IT BE FITTED?
 570  E=0
 580  FOR I=1 TO LEN(W$(W1))
 590  S$=MID$(W$(W1),I,1)
 600  REM**OFF THE EDGE?
 610  IF R2>15 OR R2<1 OR C2>15 OR C2<1
      THEN I=LEN(W$(W1)):E=1:GOTO 690
 620  IF S$(R2,C2)<>"" AND S$(R2,C2)<>S$
      THEN I=LEN(W$(W1)):E=1:GOTO 690
 630  R1=R2:C1=C2:R2(I)=R2:C2(I)=C2
 640  ON D(D) GOSUB 1250, 1260, 1270,
      1280, 1300, 1310, 1320, 1330
 650  NEXT I
 660  REM**OK WE HAVE A WORDFIT SO PUT
      IT IN THE ARRAY
 670  FOR I=1 TO LEN(W$(W1))
 680  S$(R2(I),C2(I))=MID$(W$(W1),I,1)
 690  NEXT I
 700  IF E=1 THEN E=0:GOTO 720
 710  D=8:E=1
 720  NEXT D
 730  IF E=1 THEN C=15:R=15:GOTO 760
 740  REM**MUST HAVE FAILED TO FIND FIT
      SO TRY ELSEWHERE
 750  PRINT "HARD!"
 760  NEXT C
 770  NEXT R
 780  IF E=1 THEN E=0:GOTO 870
 790  REM**TO GET HERE MUST HAVE FAILED
      FOR ENTIRE ARRAY
 800  CLS:PRINT:PRINT "SORRY, I CANT
      COPE WITH' "
 810  PRINT W$(W1);"DO YOU  WANT TO START "
 820  PRINT"AGAIN (PRESS 'S') OR PRINT OUT THE"
 830  PRINT" PARTLY DONE WORDSQUARE
      ANYWAY (PRESS 'P')"
 840  Q$=GET$
 850  IF Q$="S" THEN RUN ELSE CLS: GOTO 900
 860  REM**GET THE NEXT WORD
 870  NEXT W1
 880  REM**PRINT OUT THE ARRAY TO SCREEN
 890  PRINT "GOT IT!!!!";:FOR A=1 TO
      2000:NEXT:CLS
 900  FOR R=1 TO 15
 910  PRINT
 920  FOR C=1 TO 15
 930  IF S$(R,C)="" THEN PRINT". ";
      ELSE PRINT S$(R,C);" ";
 940  NEXT C
 950  NEXT R
 960  PRINT:PRINT:PRINT "PRESS A KEY";
 970  Q$=GET$
 980  CLS
 990  PRINT:PRINT"TYPE IN A TITLE FOR
      YOUR WORDSQUARE"
1000  PRINT "THEN PRESS RETURN"
1010  INPUT T$
1020  PRINT
1030  PRINT "DO YOU WANT THE WORDLIST
      PRINTED ?";
1040  Q$=GET$
1050  IF Q$<>"Y" AND Q$<>"N" THEN 1040
1060  PRINT:PRINT:PRINT TAB(15);T$
1070  PRINT
1080  FOR R=1 TO 15
1090  FOR C=1 TO 15
1100  IF S$(R,C)="" THEN PRINT
      CHR$(RND(26)+64);" ";ELSE PRINT S$(R,C);
      " ";
1110  NEXT C
1120  IF Q$="N" THEN PRINT TAB(30);
      STRING$(LEN(W$(R)),"-") ELSE PRINT
      TAB(30);W$(R)
1130  NEXT R
1140  IF W<>16 THEN END
1150  IF Q$="N" THEN PRINT TAB(30);
      STRING$(LEN(W$(W)),"-") ELSE PRINT
      TAB(30);W$(W)
1160  END
1170  REM**INPUT TESTING
1180  IF LEN(W$(W))>15 THEN S=1:
      PRINT"THIS WORD IS TOO LONG, TRY AGAIN":
      RETURN
1190  FOR A=1 TO LEN(W$(W))
1200  S$=MID$(W$(W),A,1)
1210  IF S$<"A" OR S$>"Z" THEN S=1:
      A=LEN(W$(W)):
1220  IF S=1 THEN PRINT "LETTERS ONLY PLEASE"
1230  RETURN
1240  REM**HORIZONTALS AND VERTICALS
1250  R2=R1+1:C2=C1:RETURN
1260  R2=R1-1:C2=C1:RETURN
1270  C2=C1+1:R2=R1:RETURN
1280  C2=C1-1:R2=R1:RETURN
1290  REM**NOW THE DIAGONALS
1300  R2=R1+1:C2=C1+1:RETURN
1310  R2=R1+1:C2=C1-1:RETURN
1320  R2=R1-1:C2=C2+1:RETURN
1330  R2=R1-1:C2=C2-1:RETURN
1340  REM**WORDLENGTH SORT
1350  S=0
1360  FOR W1=1 TO W-1
1370  IF LEN(W$(W1))<LEN(W$(W1+1))
      S$=W$(W1):W$(W1)=W$(W1+1):W$(W1+1)=S$:
      S=1
1380  NEXT
1390  IF S=1 THEN 1350
1400  RETURN
```

Mike James

SQUISH

See how long you can keep the ball in play.

PROGRAM STRUCTURE

Statement	Function	Action
Lines 10-30	Set Up	Selects MODE 4 and calls routines to display text and to initialise variables.
Lines 40-90	Balls	Delivers five balls to player.
Lines 100-170	End Game	Line 100 calls routine used at finish of use. Line 110 calls that to display end of turn messages. Lines 120 to 170 give option of another turn and return to MODE 7 if not required.
Lines 180-320	DEFPROCTEXT	Clears screen, prints title and instructions.
Lines 330-400	DEFPROCINIT	Defines graphics character 224 as a solid square and sets background colour to red. Lines 380-390 give instantaneous repeat when a key is pressed and line 400 enables use of arrow keys. Defines array T for timing.
Lines 410-540	DEFPROCPLAY	Calls routine to display 'court', prints number of ball and releases ball from random position behind bat. Calls routine to draw bat. Times play and prints time and bleeps when ball is lost. Calls routine to move bat.
Lines 550-630	DEFPROCSCR	Displays three-sided playing area.
Lines 640-690	DEFPROCBAT	Displays moving bat.
Lines 700-790	DEFPROCBALL	Displays ball and unplots its old positions. Calls routine for sound and updates time on bounce or if ball goes out of play.
Lines 800-830	DEFPROCFIN	Restores arrow keys and repeat key to normal operation.
Lines 840-890	DEFPROCDIR	Responds to press of arrow keys by appropriate movement.
Lines 900-990	DEFPROCMES	After a short delay, clears screen and displays times taken for each ball.
Lines 1000-1030	DEFPROCBEEP	Emits bounce sound and updates time displayed.
Line 1040	Screen Memory	Gives address of point x,y on screen.

This seemingly simple game is actually trickier than you'd imagine. The object of the game is to keep returning the ball for as long as possible.

A three-sided 'court' is displayed on the screen and a 'bat' is positioned near the open side (on the left). The bat can be moved up and down using the cursor control (arrow) keys. Towards the far wall (the right-hand side) there is a 'grid'. The effect of this grid is to deflect the ball sometimes and increase the difficulty of judging where the ball will return to.

You have five balls and the time each ball spends 'on screen' is displayed while you play. At the end of your turn, the time taken for each ball is listed.

TECHNICAL DETAILS

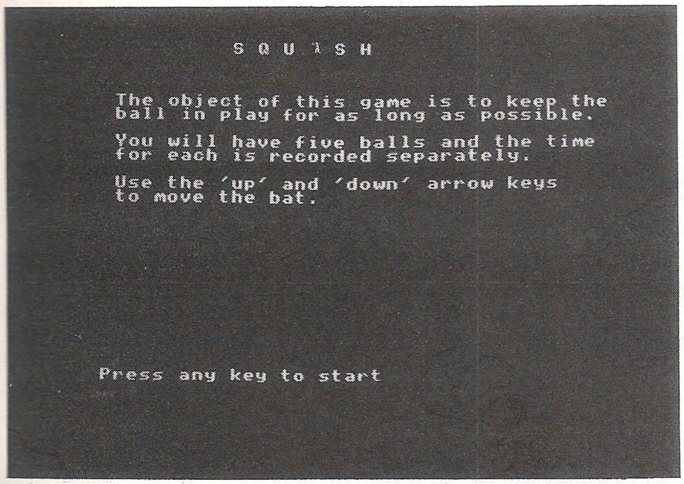
The techniques used in this game are similar to those used in 'Breakthrough'. Details of the screen mapping and keyboard repeat functions can be found there.

HOW TO PLAY

When play starts, the ball will come from a random position behind your bat, will bounce around the court and then return towards the open end. Here you must hit it or it will go out of play!

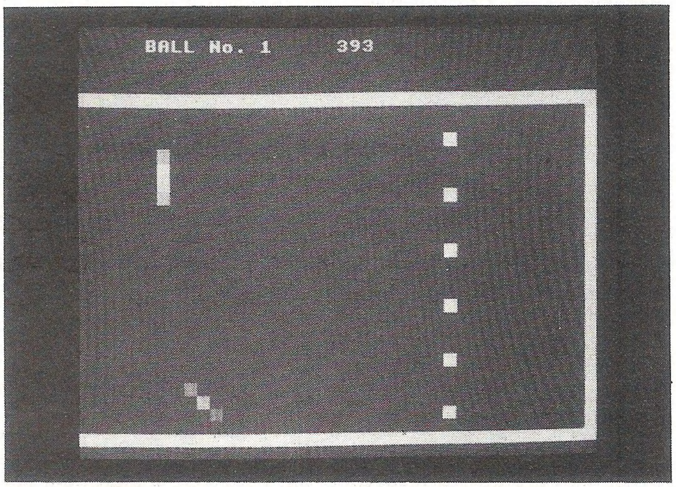
Use the UP and DOWN arrow keys to control your bat. The repeat function has been set so that the bat will continue to move all the time you keep one of these keys depressed.

At the end of each turn you have the option of another go. Typing 'NO' to this question is the only way to terminate play — if you try exiting from it by any other means, the repeat key will still be set in instantaneous mode, with disastrous results!



```
          S Q U I S H

 The object of this game is to keep the
 ball in play for as long as possible.

 You will have five balls and the time
 for each is recorded separately.

 Use the 'up' and 'down' arrow keys
 to move the bat.




Press any key to start
```

Left: The introductory page from the program.



```
     BALL No. 1      393
```

Right: The game in action, so far the player has scored 393 with his first ball.

```
  10  MODE 4
  20  PROCTEXT
  30  PROCINIT
  40  BALL=5
  50  REPEAT
  60  PROCPLAY
  70  FOR I=1 TO 1000:NEXT I
  80  BALL=BALL-1
  90  UNTIL BALL=0
 100  PROCFIN
 110  PROCMES
 120  INPUT "Another game",A$
 130  A$=LEFT$(A$,1)
 140  IF A$="Y" THEN CLEAR:GOTO 30
 150  IF A$<>"N" THEN PRINT "ANSWER Yes
      or No":GOTO 120
 160  MODE 7
 170  END
 180  DEFPROCTEXT
 190  CLS
 200  PRINT:PRINT TAB(10);"S Q U I S H"
 210  PRINT:PRINT:PRINT
 220  PRINT" The object of this game is
      to keep the"
 230  PRINT " ball in play for as long
      as possible."
 240  PRINT
 250  PRINT " You will have five balls
      and the time"
 260  PRINT " for each is recorded separately."
 270  PRINT
 280  PRINT " Use the 'up' and 'down'
      arrow keys"
 290  PRINT " to move the bat."
 300  PRINT TAB(0,25);"Press any key to start"
 310  A$=GET$
 320  ENDPROC
 330  DEFPROCINIT
 340  VDU 23,224,&FF,&FF,&FF,&FF,&FF,&FF,
      &FF,&FF
 350  VDU 19,0,1,0,0,0
 360  *FX 11 1
 370  *FX 12 1
 380  *FX 4 1
 390  DIM T(5)
 400  ENDPROC
 410  DEFPROCPLAY
 420  PROCSCR
 430  PRINT TAB(5,1);"BALL No. ";6-BALL;
 440  X=0:Y=RND(10)+10
 450  XB=6:YB=10:A=1:HV=1:VV=1
 460  E=0
 470  PROCBAT:PROCBAT:PROCBAT
 480  TIME=0
 490  PROCBAT
 500  PROCBALL
 510  IF E=1 THEN T(6-BALL)=TIME:
      SOUND 2,15,100,20:PRINT TAB(20,1);
      T(6-BALL):ENDPROC
 520  PROCDIR
 530  GOTO 490
 540  ENDPROC
 550  DEFPROCSCR
 560  CLS
 570  PRINT TAB(0,5);STRING$(40,CHR$(224));
 580  PRINT TAB(0,30);STRING$(40,CHR$(224));
 590  FOR I=5 TO 30
 600  PRINT TAB(39,I);CHR$(224);
 610  IF INT(I/4)*4=I THEN PRINT TAB(28,I);
      CHR$(224);
 620  NEXT I
 630  ENDPROC
 640  DEFPROCBAT
 650  YB=A+YB
 660  IF YB<7 THEN YB=YB+1
 670  IF YB>28 THEN YB=YB-1
 680  IF A=1 THEN PRINT TAB(XB,YB+1);CHS$(224);
      TAB(XB,YB-2);" ";ELSE PRINT TAB(XB,YB-1);
      CHR$(224);:IF YB<>28 THEN PRINT
      TAB(XB,YB-2);" ";
 690  ENDPROC
 700  DEFPROCBALL
 710  OX=X:OY=Y
 720  X=X+HV:Y=Y+VV
 730  IF X<1 THEN E=1:ENDPROC
 740  IF  X>39 THEN PROCBEEP:HV=-HV:X=X+HV
 750  IF Y<6 OR Y>29 THEN PROCBEEP:
      VV=-VV:Y=Y+VV
 760  IF ?FNS(X,Y;0)<>0 THEN PROCBEEP:
      HV=-HV:X=X+HV
 770  IF XB<>OX OR YB<>OY THEN PRINT
      TAB(OX,OY);" ";
 780  PRINT TAB(X,Y);CHR$(224);
 790  ENDPROC
 800  DEFPROCFIN
 810  *FX 4,0
 820  *FX 12,0
 830  ENDPROC
 840  DEFPROCDIR
 850  A=INKEY(0)
 860  *FX 15,0
 870  A=-(2*(A-&8A)-1)
 880  IF A<>-1 AND A<>1 THEN A=0
 890  ENDPROC
 900  DEFPROCMES
 910  FOR I=1 TO 1000:NEXT I
 920  CLS
 930  PRINT TAB(5,10);"Your times were:"
 940  PRINT
 950  FOR I=1 TO 5
 960  PRINT "BALL ";I;TAB(10);T(I)/100;
      TAB(15);"SECONDS"
 970  NEXT I
 980  PRINT:PRINT
 990  ENDPROC
1000  DEFPROCBEEP
1010  SOUND 2,-15,150,1
1020  PRINT TAB(20,1);TIME;
1030  ENDPROC
1040  DEF FNS(X,Y,N)=HIMEM+(Y*40+X)*8+N
```

Converted from a program by Ian L Powell

BATTLE OF BRITAIN

Try not to let history be re-written as you run this simulation.

This program is based on two months (August — September, 1940) of the Battle of Britain. The computer plays the German side and its objective is to cripple Fighter Command. You command the British side and your objective is to inflict enough casualties to deter further attack. The game is played over ten turns, simulating one week of action per turn. During each turn the German side launches three attacks on Britain. These raids can be aimed at:

A — Fighter Bases
B — Radar Stations
C — Industrial Centres

The British have 100 industrial factors at their disposal. These factors may be used to build fighters or radar stations; the term, radar station, also includes various ground defences. Fighters cost one industrial factor each and radar stations cost ten industrial factors each.

Industrial factors are renewed at the start of each week. However, industrial factors can be destroyed by bombing, in which case they are permanently lost, and so the total of available industrial factors will dwindle as the weeks pass.

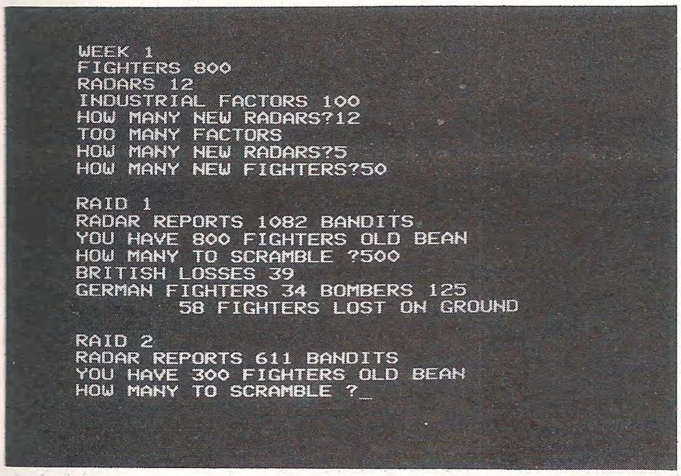
Blitzed Again

In combat, British fighters are worth more than German fighters. This is simply because the German fighters are escorting bombers, and thus have less freedom of action. German industrial factors cannot be replaced because of the small time scale, just ten weeks.

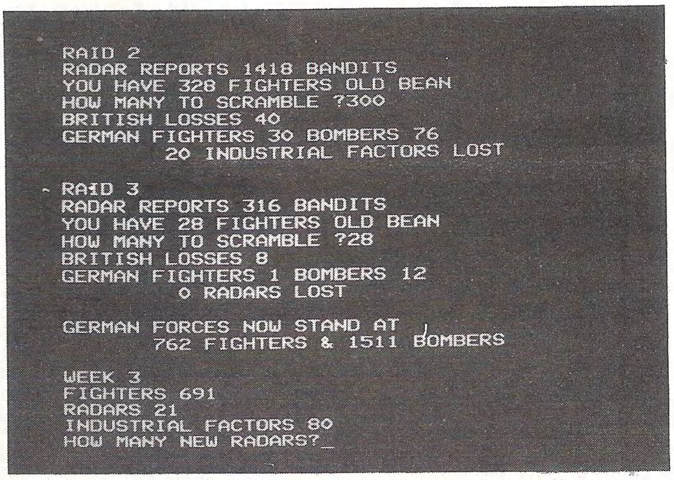
German experimental fighter-bomber Gruppes have been added to the bomber force. German 110 twin engine fighters have been ignored, except when operating as bombers.

For the purpose of this game, initial historical German tactics are maintained throughout the entire game. Historically, heavy casualties forced the Germans to operate small bomber formations with large fighter escorts from mid-September, 1940 onwards. British reinforcements of about 100 fighters per week are also fairly accurate although the lack of trained pilots (simulated by switching resources to other things) made this figure considerably lower in practice.

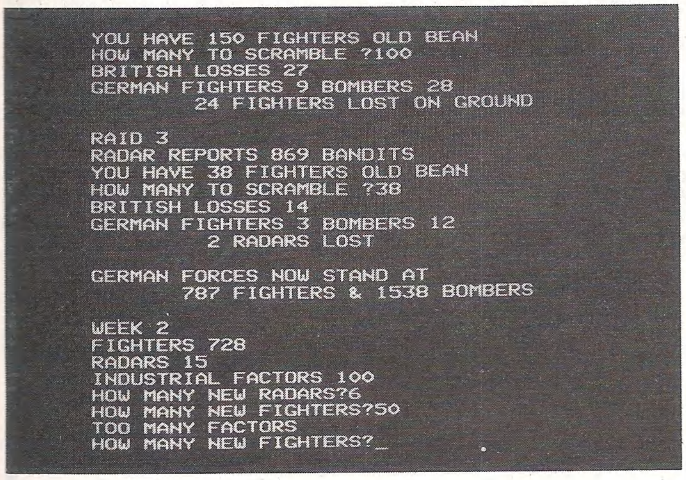
German reinforcements, 40 fighters and 125 bombers per week, are approximately accurate too; the bomber figures are 35% larger but this simulates the return of damaged



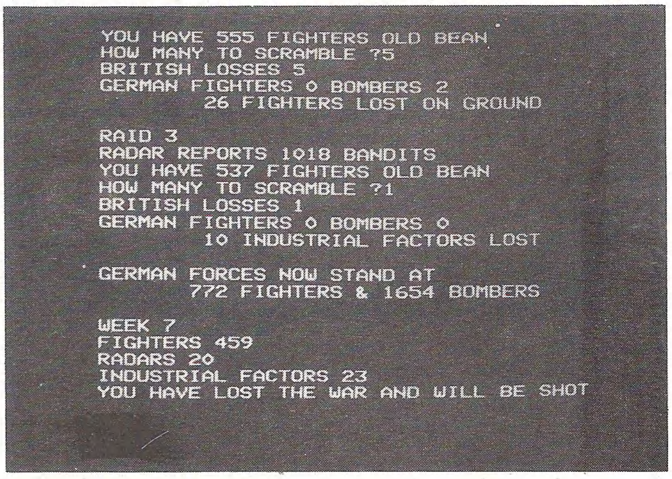
```
WEEK 1
FIGHTERS 800
RADARS 12
INDUSTRIAL FACTORS 100
HOW MANY NEW RADARS?12
TOO MANY FACTORS
HOW MANY NEW RADARS?5
HOW MANY NEW FIGHTERS?50

RAID 1
RADAR REPORTS 1082 BANDITS
YOU HAVE 800 FIGHTERS OLD BEAN
HOW MANY TO SCRAMBLE ?500
BRITISH LOSSES 39
GERMAN FIGHTERS 34 BOMBERS 125
        58 FIGHTERS LOST ON GROUND

RAID 2
RADAR REPORTS 611 BANDITS
YOU HAVE 300 FIGHTERS OLD BEAN
HOW MANY TO SCRAMBLE ?_
```



```
RAID 2
RADAR REPORTS 1418 BANDITS
YOU HAVE 328 FIGHTERS OLD BEAN
HOW MANY TO SCRAMBLE ?300
BRITISH LOSSES 40
GERMAN FIGHTERS 30 BOMBERS 76
        20 INDUSTRIAL FACTORS LOST

RAID 3
RADAR REPORTS 316 BANDITS
YOU HAVE 28 FIGHTERS OLD BEAN
HOW MANY TO SCRAMBLE ?28
BRITISH LOSSES 8
GERMAN FIGHTERS 1 BOMBERS 12
        0 RADARS LOST

GERMAN FORCES NOW STAND AT
        762 FIGHTERS & 1511 BOMBERS

WEEK 3
FIGHTERS 691
RADARS 21
INDUSTRIAL FACTORS 80
HOW MANY NEW RADARS?_
```



```
YOU HAVE 150 FIGHTERS OLD BEAN
HOW MANY TO SCRAMBLE ?100
BRITISH LOSSES 27
GERMAN FIGHTERS 9 BOMBERS 28
        24 FIGHTERS LOST ON GROUND

RAID 3
RADAR REPORTS 869 BANDITS
YOU HAVE 38 FIGHTERS OLD BEAN
HOW MANY TO SCRAMBLE ?38
BRITISH LOSSES 14
GERMAN FIGHTERS 3 BOMBERS 12
        2 RADARS LOST

GERMAN FORCES NOW STAND AT
        787 FIGHTERS & 1538 BOMBERS

WEEK 2
FIGHTERS 728
RADARS 15
INDUSTRIAL FACTORS 100
HOW MANY NEW RADARS?6
HOW MANY NEW FIGHTERS?50
TOO MANY FACTORS
HOW MANY NEW FIGHTERS?_
```



```
YOU HAVE 555 FIGHTERS OLD BEAN
HOW MANY TO SCRAMBLE ?5
BRITISH LOSSES 5
GERMAN FIGHTERS 0 BOMBERS 2
        26 FIGHTERS LOST ON GROUND

RAID 3
RADAR REPORTS 1018 BANDITS
YOU HAVE 537 FIGHTERS OLD BEAN
HOW MANY TO SCRAMBLE ?1
BRITISH LOSSES 1
GERMAN FIGHTERS 0 BOMBERS 0
        10 INDUSTRIAL FACTORS LOST

GERMAN FORCES NOW STAND AT
        772 FIGHTERS & 1654 BOMBERS

WEEK 7
FIGHTERS 459
RADARS 20
INDUSTRIAL FACTORS 23
YOU HAVE LOST THE WAR AND WILL BE SHOT
```

bombers to field strength.

Initial game casualities will be very heavy. Roughly half of the German bomber casualties will be regarded as damaged only.

Victory Rolls OK

If the Germans have fallen below 1,000 bombers at the start of a new move, they will give up the attack and the British will win. You will be a hero! However, if the British drop below 500 fighters at the start of a new move, they will be invaded and crushed. You will be shot!!

If at the end of the game, the British have more points value left than the Germans, the British win having staved off invasion. You will be given a knighthood. If at the end of the game, the Germans have more points left than the British (bombers count 1, fighters count 3) then the Germans will continue their attack, but you won't be there to stop them.

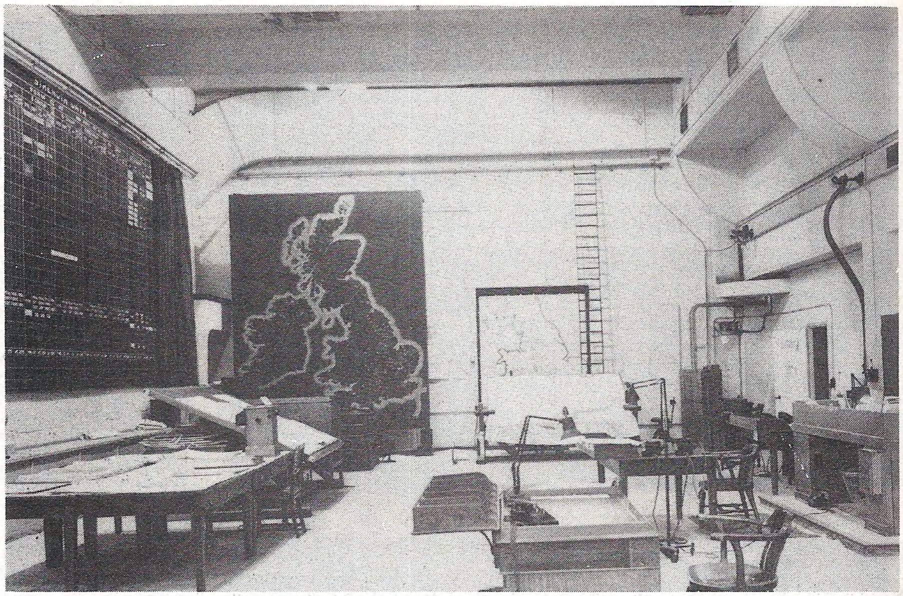
PROGRAM STRUCTURE

Statement	Function	Action
Lines 10-20	Set Up	Selects MODE 7 and sets initial values of the variables.
Lines 30-80	Print	Prints recurrent variables at beginning of week.
Lines 90-110	End Game	Checks whether you 'won' or 'lost' and prints appropriate message.
Lines 120-150	Input	Requests input for variables and introduces random element.
Lines 160-270	Calculation	Calculates variables and introduces random element.
Lines 280-340	Messages	Reports on state of play.
Lines 350-480	Raid	Works out the effects of raid and fighter attack.
Lines 490-500	Raid End	Reports results of raid.
Lines 510-730	Losses	Calculates 'extra' losses and prints appropriate message.
Lines 740-760	Next Week	Reports condition of opposing forces at beginning of subsequent weeks.
Lines 770-860	End Game	Messages for indeterminate ending.
Line 870	Factors	Prints message if you use more factors than there are available.

BATTLE OF BRITAIN

HOW TO PLAY

You have to decide how many of your fighters to scramble on each raid — and during any week, each of your aircraft can only be used in one of the three raids. You also have to decide how to use your quota of one hundred industrial factors; you can use them on either radar (each of which will cost you ten factors) or on new fighters (which only cost one factor each). If you try to use more factors than you have, the computer will let you know and allow you to adjust your plans. The computer will prompt for all the decisions you have to make.



```
10   MODE 7
20   A=800: B=12: C=100: D=800: E=1600
30   FOR W=1 TO 10
40   PRINT
50   PRINT "WEEK ";W
60   PRINT "FIGHTERS ";A
70   PRINT "RADARS ";B
80   PRINT "INDUSTRIAL FACTORS ";C
90   IF A < 500 THEN PRINT "YOU HAVE
     LOST THE WAR AND WILL BE SHOT"
100  IF E < 1000 THEN PRINT "YOU HAVE
     WON THE BATTLE AND ARE A HERO"
110  IF A<500 OR E<1000 THEN PRINT:END
120  INPUT "HOW MANY NEW RADARS",P
130  IF C<10*P THEN GOSUB 870:GOTO 120
140  INPUT "HOW MANY NEW FIGHTERS",K
150  IF (C-10*P)<K THEN GOSUB 870:GOTO 140
160  M=A:N=D:O=E
170  FOR R=1 TO 3
180  IF R=1 THEN Q=RND(2)
190  IF R=2 THEN Q=RND(3)
200  IF R=3 THEN Q=RND(2)+1
210  IF R=1 THEN G=RND(INT(N/2)):
     F=RND(INT(O/2))
220  IF R=2 THEN G=RND(N-1):F=RND(O-1)
230  IF R=3 THEN G=N:F=O
240  N=N-G:O=O-F
250  X=13-B:IF X<1 THEN X=1
260  Z=INT(((G+F)*X)/10)
270  X=RND(2*Z)-Z
280  PRINT
290  PRINT "RAID ";R
300  IF B<1 THEN PRINT "YOUR RADAR IS
     DEAD":GOTO 320
310  PRINT "RADAR REPORTS ";ABS(G+F+X);
     " BANDITS"
320  PRINT "YOU HAVE ";M;" FIGHTERS
     OLD BEAN":GOTO 330
330  INPUT "HOW MANY TO SCRAMBLE ",H
340  IF H>M THEN PRINT "TOO MANY
     FIGHTERS OLD BEAN ":GOTO 330
350  M=M-H
360  S=5*H:T=G*3+F
370  IF ABS(S-T)*5>(S+T) THEN GOTO 410
380  X=RND(3)+8:U=INT((T+INT(X/2))/X)
390  X=RND(3)+8:V=INT((S+INT(X/2))/X)
400  GOTO 450
410  IF S<T THEN GOTO 440
420  U=INT(T/9):V=INT(S/12)
430  GOTO 450
440  U=INT(T/12):V=INT(S/9)
450  L=INT(U/5):I=0:J=0:IF V>0 THEN
     I=INT((RND(V)+V)/9):J=V-I*3
460  IF L>H THEN L=H
470  IF I>G THEN I=G
480  IF J>F THEN J=F
490  PRINT "BRITISH LOSSES ";L
500  PRINT "GERMAN FIGHTERS ";I;"
     BOMBERS ";J
510  IF S>=T THEN X=8
520  IF S<T THEN X=12
530  IF S<2*T THEN X=18
540  X=(F-J)*X
550  A=A-L:D=D-I:E=E-J
560  IF Q=2 THEN GOTO 640
570  IF Q=3 THEN GOTO 680
580  X=INT((X-100)/200):IF X>A  THEN X=A
590  IF X<0 THEN X=0
600  IF R=2 THEN M=M-INT(X/2)
610  IF M<0 THEN M=0
620  PRINT X;" FIGHTERS LOST ON GROUND":A=A-X
630  GOTO 710
640  X=INT((X-2000)/4000):IF X>B THEN X=B
650  IF X<0 THEN X=0
660  PRINT X;" RADARS LOST ":B=B-X
670  GOTO 710
680  X=INT((X-400)/800):IF X>C THEN X=C
690  IF X<0 THEN X=0
700  PRINT X;" INDUSTRIAL FACTORS LOST":C=C-X
710  NEXT R:PRINT
720  A=A+K:B=B+P
730  D=D+40:E=E+125
740  PRINT"GERMAN FORCES NOW STAND AT "
750  PRINT D;" FIGHTERS & ";E;" BOMBERS"
760  NEXT W
770  IF A*5>D*3+E THEN GOTO 830
780  PRINT "THE GERMANS WILL CONTINUE
     THE ATTACK"
790  PRINT "YOU ARE NOW THE AMBASSADOR
     OF KALAMAZOO"
800  PRINT "DON'T COME BACK"
810  PRINT
820  END
830  PRINT "YOU HAVE STAVED OFF INVASION"
840  PRINT "YOU HAVE BEEN KNIGHTED"
850  PRINT
860  END
870  PRINT "TOO MANY FACTORS":RETURN
```

ZX COMPUTING

Britain's Biggest Magazine For The Sinclair User

Over 100 pages of information and programs for the ZX81 and ZX80 user including a 1K Chess routine!

ZX Computing is a quarterly magazine with over 100 pages of programs, articles, hints and tips for the ZX81 and ZX80. We've put the magazine together with just one aim in mind — to make sure you get the most use and enjoyment out of your Sinclair computer.

In the first issue, as well as a host of 1K and 16K programs (the majority dumped directly onto the printer), we've got two articles to help dispel the mysteries of machine code, a chess routine which takes up just 1K, an explanation of PEEK and POKE, an entire section on business uses of the ZX81 (complete with a program to handle the accounts of up to 100 customers) and a construction article which will show you — even if you've never used a soldering iron before — how to cheaply double the memory on your ZX81.

If you're serious about exploring the full potential of your Sinclair computer, and you want an easy-to-understand series of articles to improve and develop your programming skills, then ZX Computing is for you. Issue one is on sale now.

- **Software Reviews** — which programs are the best buy?
- **Business Routines** — put your ZX to work!
- **Expansion Systems** — how good and how much?
- **Machine Code for ZX81**— secrets revealed at last!
- **DIY Memory Upgrade** — cheap way of adding bytes

SUBSCRIPTION ORDER FORM

Cut out and SEND TO : ZX COMPUTING

513 LONDON ROAD,
THORNTON HEATH,
SURREY,
CR4 6AR.

POST FREE SUBSCRIPTION

Please use BLOCK CAPITALS and include post codes.

Name (Mr/Mrs/Miss) ....
delete accordingly
Address ....
....
....
Signature ....
Date ....

82 Summer

Please commence my subscription to ZX Computing with the ....issue.

SUBSCRIPTION RATES (tick ☐ as appropriate)

£7.00 for 4 issues UK ☐

£1.75 for a single copy of the ......issue ☐

I am enclosing my (delete as necessary) Cheque/Postal Order/International Money Order for £..........
(made payable to ASP Ltd)
OR
*Debit my Access/Barclaycard **
*(*delete as necessary)*

Insert card no. ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Converted from a program by Paul Williams

FOX AND HOUNDS

No need to get 'blooded' here, saddle up and join the humane hunters.

Photograph courtesy of Rank Film Distributors

Fox and Hounds is a game of strategy played on a chess board grid displayed on the computer's VDU. The sole object of the game is to trap the fox which is automatically operated by the computer, with the pieces you control, the hounds. The fox starts from the top of the board and the hounds at the bottom. If the fox reaches the bottom row of the board before being trapped, it has 'won'.

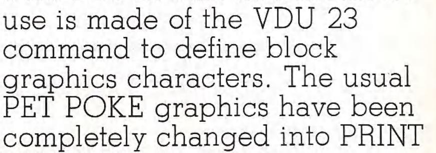
The fox can move both forwards and backwards, like a bishop in chess, the hounds may move only forwards. All the pieces on the board move diagonally by one square at a time.

Tally Ho!

You can win by surrounding the fox with the hounds so that it cannot move or you can trap it into a corner or against the side of the board. The algorithm used by the computer is fairly ruthless — it will not step into any traps it can spot.

TECHNICAL DETAILS

The program was converted from PET BASIC and extensive use is made of the VDU 23 command to define block graphics characters. The usual PET POKE graphics have been completely changed into PRINT TAB(X,Y) commands.



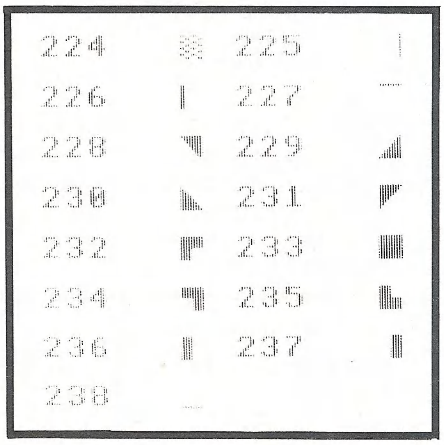
If you have the Model B one possible suggestion is to use MODE 1 and colour the Fox and Hounds differently, it just adds variety!

PROGRAM STRUCTURE

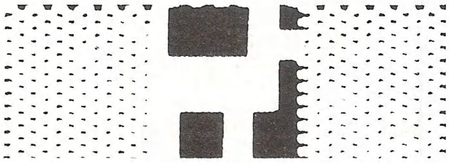
Statement	Function	Action
Lines 10-20	Set Mode	Selects MODE 4 and sets foreground colour to yellow.
Lines 30-170	Characters	Defines graphics characters (see Table 1).
Lines 180-240	Start	Prints title and asks for player to decide how many hounds are required.
Lines 250-450	Set Board	Displays board, and text to the right of it.
Lines 460-540	Pieces	Calls subroutines to set up pieces in position and display them.
Lines 550-920	Move Logic	Computes move logic. If move is legal it will be implemented. If not permitted, a 'bleep' sounds.
Lines 930-940	Patterns	Creates the 'fox', the 'hounds' and the blank spaces by combinations of the characters defined at the beginning of program.
Lines 950-990	Positions	Routine to print pieces at the required positions on the board.
Lines 1000-1030	End Game	Restarts game if required.
Lines 1040-1070	Move Logic	Additional logic, relating to earlier section.
Lines 1080-1090	End Game	Message displayed if computer wins.

GRAPHICS CHARACTERS

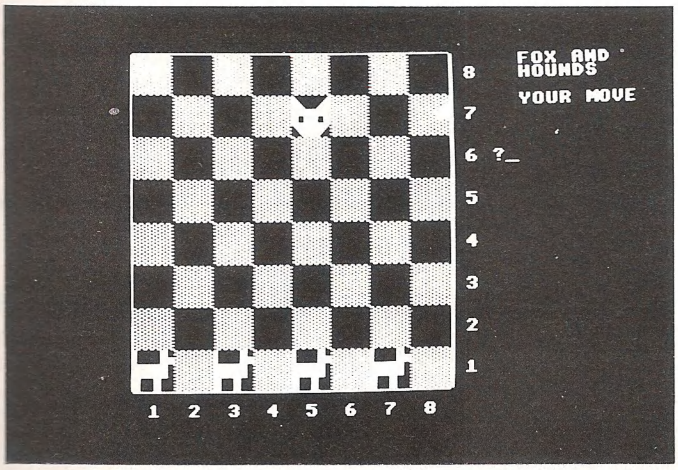
The following table shows the graphics characters used to create the fox and hounds.



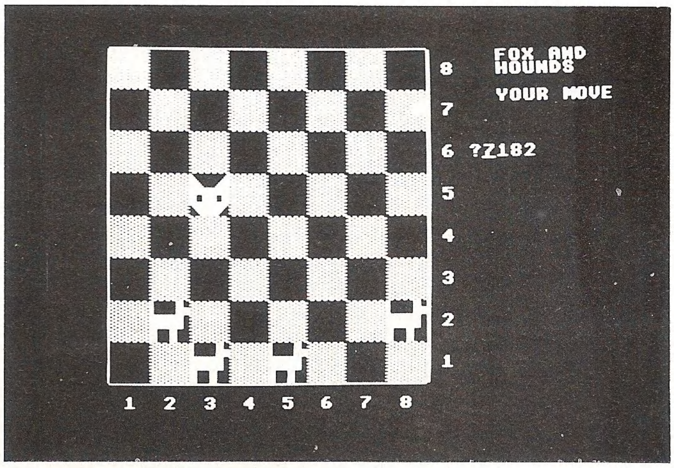
224	▒	225	▕
226	▏	227	‾
228	◥	229	◢
230	◣	231	◤
232	▛	233	▮
234	▜	235	▙
236	‖	237	‖
238	_		



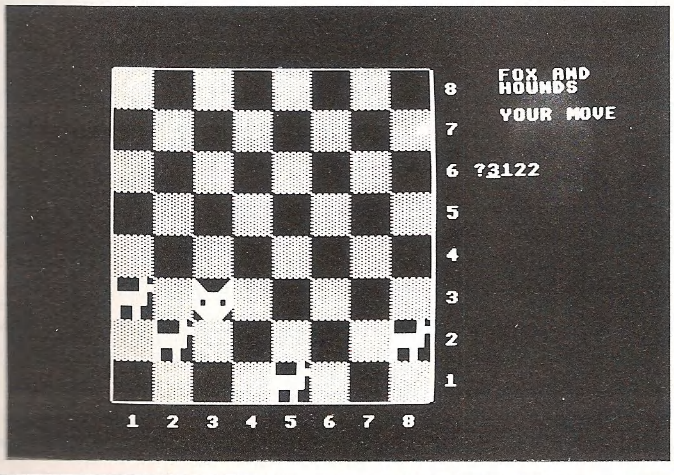
The screen photographs below show, clockwise from top left, a game in progress. If the fox moves to co-ordinate 4,2 the game will be lost as the hounds cannot move backwards.



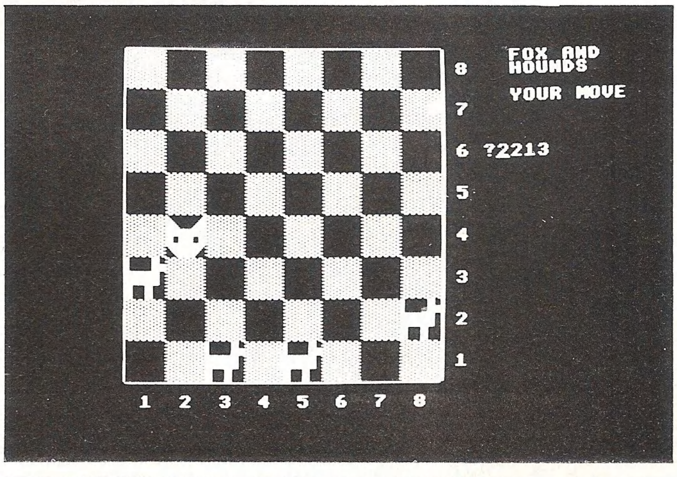
```
FOX AND
HOUNDS
YOUR MOVE
?_
```



```
FOX AND
HOUNDS
YOUR MOVE
?7182
```



```
FOX AND
HOUNDS
YOUR MOVE
?3122
```



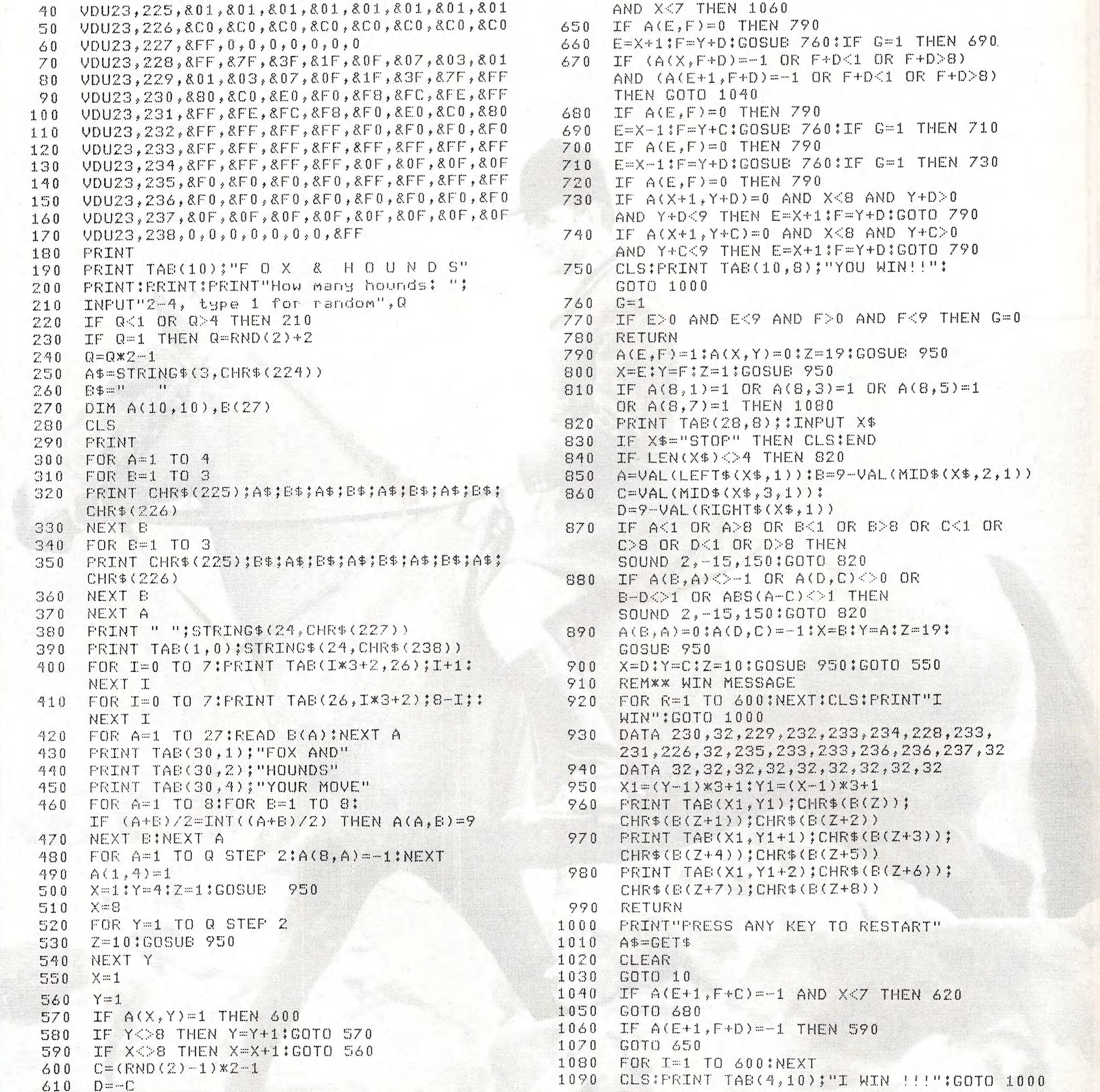
```
FOX AND
HOUNDS
YOUR MOVE
?2213
```

HOW TO PLAY

After the board has been constructed on the screen you must wait for the cursor to appear in the top right-hand corner. At this point the program is waiting for you to input a set of co-ordinates which relate to the piece you wish to move. The input string is expected to be a four character code with the first two characters being the current square reference and the second two being the destination square reference. The input is terminated by pressing Return and the computer will then move your hound to the selected square. The computer checks all entered moves for validity and if the move requested is illegal, the input string is deleted and you must try again.

As an example of the move input we could input the string 1122. This indicates that we wish to move from the bottom left-hand square to the square with co-ordinates 2,2. All the moves are indexed from the bottom left-hand corner, this being 1,1. The co-ordinates are taken along and then up; that is you quote the column number first, followed by the row number to select a square.

The game may be terminated at any point by typing 'STOP'.

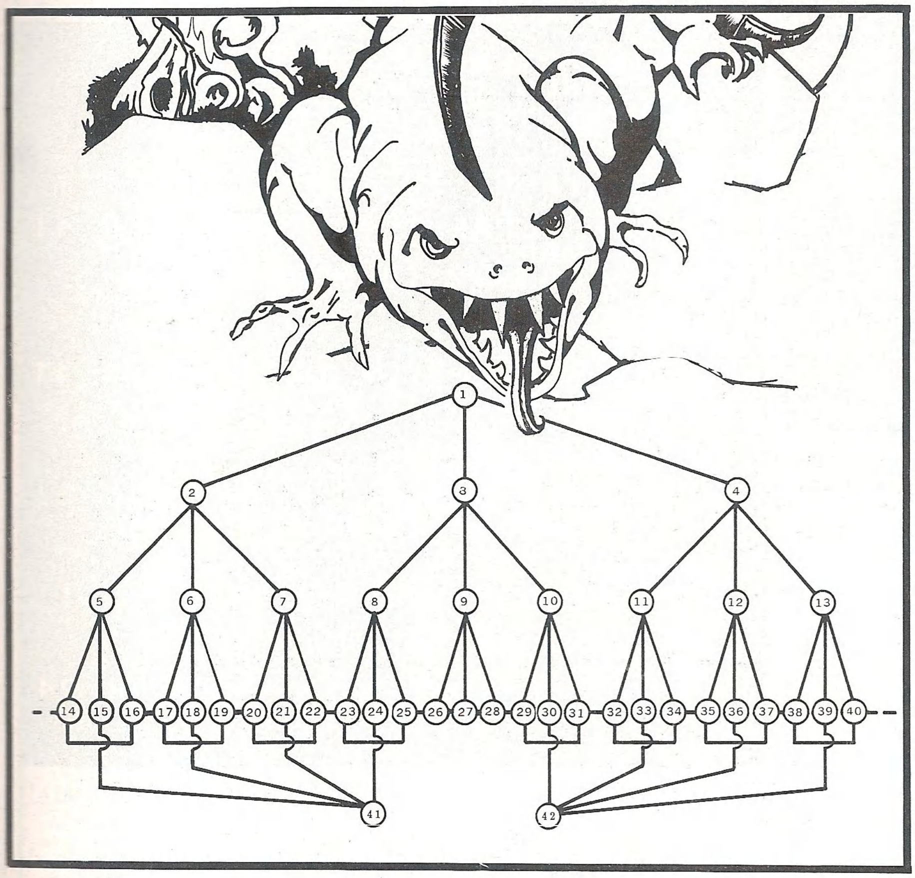


```
  10  MODE 4
  20  VDU19,1,3,0,0,0
  30  VDU23,224,&CC,&33,&CC,&33,&CC,&33,&CC,&33
  40  VDU23,225,&01,&01,&01,&01,&01,&01,&01,&01
  50  VDU23,226,&C0,&C0,&C0,&C0,&C0,&C0,&C0,&C0
  60  VDU23,227,&FF,0,0,0,0,0,0,0
  70  VDU23,228,&FF,&7F,&3F,&1F,&0F,&07,&03,&01
  80  VDU23,229,&01,&03,&07,&0F,&1F,&3F,&7F,&FF
  90  VDU23,230,&80,&C0,&E0,&F0,&F8,&FC,&FE,&FF
 100  VDU23,231,&FF,&FE,&FC,&F8,&F0,&E0,&C0,&80
 110  VDU23,232,&FF,&FF,&FF,&FF,&F0,&F0,&F0,&F0
 120  VDU23,233,&FF,&FF,&FF,&FF,&FF,&FF,&FF,&FF
 130  VDU23,234,&FF,&FF,&FF,&FF,&0F,&0F,&0F,&0F
 140  VDU23,235,&F0,&F0,&F0,&F0,&FF,&FF,&FF,&FF
 150  VDU23,236,&F0,&F0,&F0,&F0,&F0,&F0,&F0,&F0
 160  VDU23,237,&0F,&0F,&0F,&0F,&0F,&0F,&0F,&0F
 170  VDU23,238,0,0,0,0,0,0,0,&FF
 180  PRINT
 190  PRINT TAB(10);"F O X  &  H O U N D S"
 200  PRINT:RRINT:PRINT"How many hounds: ";
 210  INPUT"2-4, type 1 for random",Q
 220  IF Q<1 OR Q>4 THEN 210
 230  IF Q=1 THEN Q=RND(2)+2
 240  Q=Q*2-1
 250  A$=STRING$(3,CHR$(224))
 260  B$="   "
 270  DIM A(10,10),B(27)
 280  CLS
 290  PRINT
 300  FOR A=1 TO 4
 310  FOR B=1 TO 3
 320  PRINT CHR$(225);A$;B$;A$;B$;A$;B$;A$;B$;
      CHR$(226)
 330  NEXT B
 340  FOR B=1 TO 3
 350  PRINT CHR$(225);B$;A$;B$;A$;B$;A$;B$;A$;
      CHR$(226)
 360  NEXT B
 370  NEXT A
 380  PRINT " ";STRING$(24,CHR$(227))
 390  PRINT TAB(1,0);STRING$(24,CHR$(238))
 400  FOR I=0 TO 7:PRINT TAB(I*3+2,26);I+1:
      NEXT I
 410  FOR I=0 TO 7:PRINT TAB(26,I*3+2);8-I;:
      NEXT I
 420  FOR A=1 TO 27:READ B(A):NEXT A
 430  PRINT TAB(30,1);"FOX AND"
 440  PRINT TAB(30,2);"HOUNDS"
 450  PRINT TAB(30,4);"YOUR MOVE"
 460  FOR A=1 TO 8:FOR B=1 TO 8:
      IF (A+B)/2=INT((A+B)/2) THEN A(A,B)=9
 470  NEXT B:NEXT A
 480  FOR A=1 TO Q STEP 2:A(8,A)=-1:NEXT
 490  A(1,4)=1
 500  X=1:Y=4:Z=1:GOSUB  950
 510  X=8
 520  FOR Y=1 TO Q STEP 2
 530  Z=10:GOSUB 950
 540  NEXT Y
 550  X=1
 560  Y=1
 570  IF A(X,Y)=1 THEN 600
 580  IF Y<>8 THEN Y=Y+1:GOTO 570
 590  IF X<>8 THEN X=X+1:GOTO 560
 600  C=(RND(2)-1)*2-1
 610  D=-C
 620  E=X+1:F=Y+C:GOSUB 760
 630  IF G=1 THEN 660
 640  IF (A(X,F+C)=-1 OR F+C<1 OR F+C>8)
      AND (A(E+1,F+C)=-1 OR F+C<1 OR F+C>8)
      AND X<7 THEN 1060
 650  IF A(E,F)=0 THEN 790
 660  E=X+1:F=Y+D:GOSUB 760:IF G=1 THEN 690
 670  IF (A(X,F+D)=-1 OR F+D<1 OR F+D>8)
      AND (A(E+1,F+D)=-1 OR F+D<1 OR F+D>8)
      THEN GOTO 1040
 680  IF A(E,F)=0 THEN 790
 690  E=X-1:F=Y+C:GOSUB 760:IF G=1 THEN 710
 700  IF A(E,F)=0 THEN 790
 710  E=X-1:F=Y+D:GOSUB 760:IF G=1 THEN 730
 720  IF A(E,F)=0 THEN 790
 730  IF A(X+1,Y+D)=0 AND X<8 AND Y+D>0
      AND Y+D<9 THEN E=X+1:F=Y+D:GOTO 790
 740  IF A(X+1,Y+C)=0 AND X<8 AND Y+C>0
      AND Y+C<9 THEN E=X+1:F=Y+D:GOTO 790
 750  CLS:PRINT TAB(10,8);"YOU WIN!!":
      GOTO 1000
 760  G=1
 770  IF E>0 AND E<9 AND F>0 AND F<9 THEN G=0
 780  RETURN
 790  A(E,F)=1:A(X,Y)=0:Z=19:GOSUB 950
 800  X=E:Y=F:Z=1:GOSUB 950
 810  IF A(8,1)=1 OR A(8,3)=1 OR A(8,5)=1
      OR A(8,7)=1 THEN 1080
 820  PRINT TAB(28,8);:INPUT X$
 830  IF X$="STOP" THEN CLS:END
 840  IF LEN(X$)<>4 THEN 820
 850  A=VAL(LEFT$(X$,1)):B=9-VAL(MID$(X$,2,1))
 860  C=VAL(MID$(X$,3,1)):
      D=9-VAL(RIGHT$(X$,1))
 870  IF A<1 OR A>8 OR B<1 OR B>8 OR C<1 OR
      C>8 OR D<1 OR D>8 THEN
      SOUND 2,-15,150:GOTO 820
 880  IF A(B,A)<>-1 OR A(D,C)<>0 OR
      B-D<>1 OR ABS(A-C)<>1 THEN
      SOUND 2,-15,150:GOTO 820
 890  A(B,A)=0:A(D,C)=-1:X=B:Y=A:Z=19:
      GOSUB 950
 900  X=D:Y=C:Z=10:GOSUB 950:GOTO 550
 910  REM** WIN MESSAGE
 920  FOR R=1 TO 600:NEXT:CLS:PRINT"I
      WIN":GOTO 1000
 930  DATA 230,32,229,232,233,234,228,233,
      231,226,32,235,233,233,236,236,237,32
 940  DATA 32,32,32,32,32,32,32,32,32
 950  X1=(Y-1)*3+1:Y1=(X-1)*3+1
 960  PRINT TAB(X1,Y1);CHR$(B(Z));
      CHR$(B(Z+1));CHR$(B(Z+2))
 970  PRINT TAB(X1,Y1+1);CHR$(B(Z+3));
      CHR$(B(Z+4));CHR$(B(Z+5))
 980  PRINT TAB(X1,Y1+2);CHR$(B(Z+6));
      CHR$(B(Z+7));CHR$(B(Z+8))
 990  RETURN
1000  PRINT"PRESS ANY KEY TO RESTART"
1010  A$=GET$
1020  CLEAR
1030  GOTO 10
1040  IF A(E+1,F+C)=-1 AND X<7 THEN 620
1050  GOTO 680
1060  IF A(E+1,F+D)=-1 THEN 590
1070  GOTO 650
1080  FOR I=1 TO 600:NEXT
1090  CLS:PRINT TAB(4,10);"I WIN !!!":GOTO 1000
```

Converted from a program by Andrew Thomas and Gareth Callister

THE WHITE BARROWS

Match your magic against the evil sorcerer in this logical game of skill.



As in all good adventure games, your task is to recover the magic chalice which has been stolen by an evil sorcerer hiding in the labyrinth of the White Barrows. This labyrinth is arranged in a stable pattern (see Fig. 1) and each of the chambers has an identifying number. The sorcerer is powerless to escape from the labyrinth while carrying the chalice, but if you trap him he will drop it and promptly disappear. To aid you in your task you are equipped with a Magic Staff and a limited supply of magic.

Ghouls Gallery

Lurking in the labyrinth are dragons, serpents, dwarfs, goblins and magic spots. These remain in their own chamber in the labyrinth, they cannot move around like the sorcerer. The program will issue warnings as to the content of the chamber you are about to enter, but these are not in any particular sequence. You may receive a warning about a serpent while you are in, for example, chamber two but you will not know — unless you can remember — in which of the four possible move positions it is. You can use the VIEW command to check the chamber but this will use some of your limited stocks of magic.

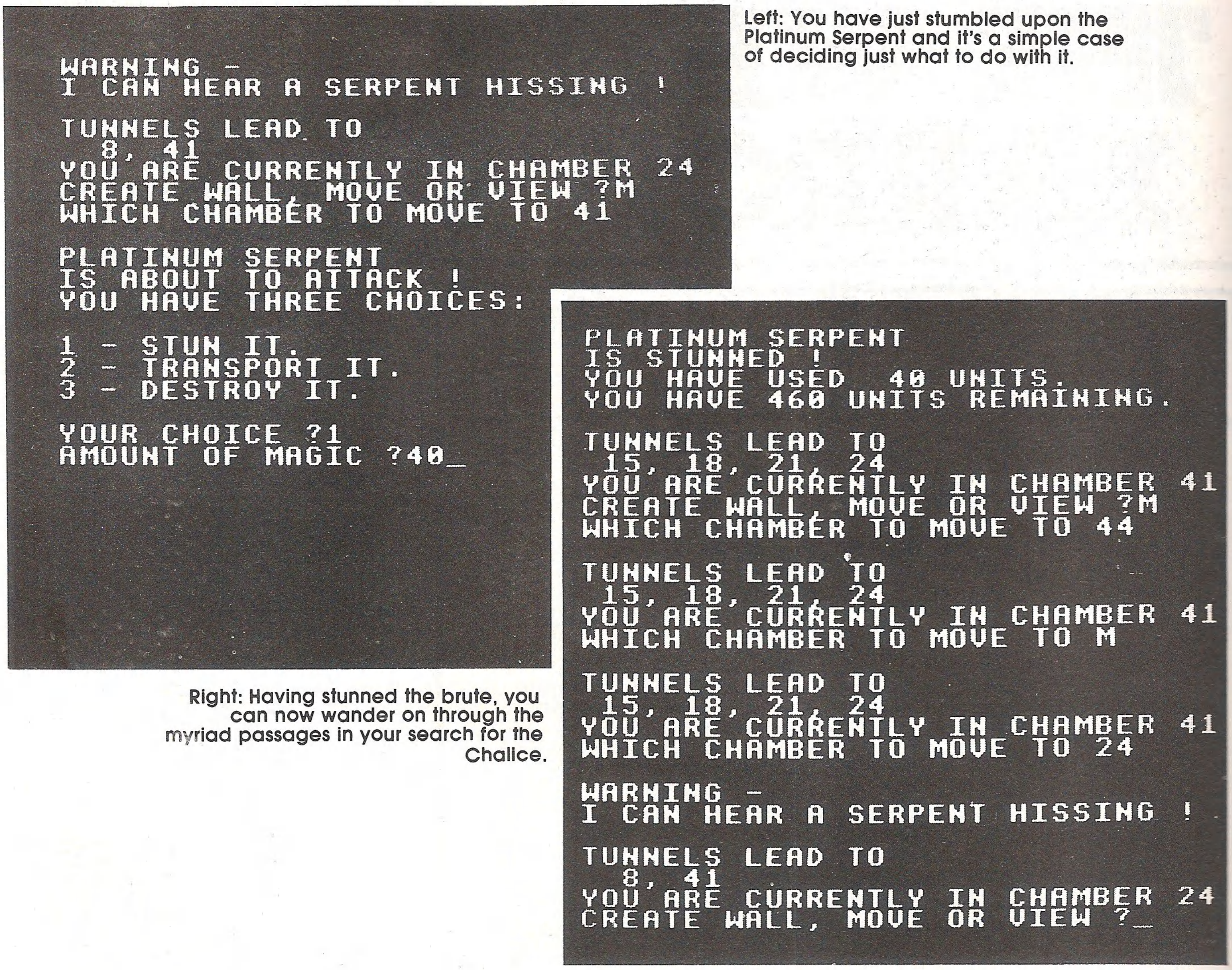
If you are unfortunate enough to run into a dragon, serpent or dwarf you are faced with one of three choices:
You can **stun** it, but you must leave the chamber immediately or it will recover and attack you.
You can **transport** it, but if you are really unlucky it may reappear in the same chamber.
You can **destroy** it, in which case the creature is removed from the game.
All the above options use up some of your magic; the actual amount depends on the creature type and which of the options you choose (these are listed in ascending order of magic required). If you do not use sufficient magic the creature may be stunned (in which case you get a second chance) or it may just get angry and attack you.

The goblins are actually on your side — but for a price. They will tell you where the sorcerer is hiding provided you supply them with gold coins which you can make with your magical ability. But don't be too quick to part with your coins, the system gives the same warning for a goblin approaching as for the friendly gnome who doesn't need bribing.

There is one further obstacle hidden in the labyrinth — a magic spot. This, if trodden on, will instantly transport you to another location within the barrow. Murphy's law determines that no matter what peril you were in before you were transported, the situation you find yourself in afterwards will be ten times worse!

A Sorcerer Full Of Secrets

The evil sorcerer does have a degree of intelligence. If you walk into the chamber in which he is hiding, he will run off. The distance he will run can be up to three chambers and, if he can find it, he will always tread on the magic spot to further confuse the issue. To retrieve the chalice you must trap him. ▶



Left: You have just stumbled upon the Platinum Serpent and it's a simple case of deciding just what to do with it.

```
WARNING -
I CAN HEAR A SERPENT HISSING !

TUNNELS LEAD TO
  8, 41
YOU ARE CURRENTLY IN CHAMBER 24
CREATE WALL, MOVE OR VIEW ?M
WHICH CHAMBER TO MOVE TO 41

PLATINUM SERPENT
IS ABOUT TO ATTACK !
YOU HAVE THREE CHOICES:

1 - STUN IT.
2 - TRANSPORT IT.
3 - DESTROY IT.

YOUR CHOICE ?1
AMOUNT OF MAGIC ?40_
```

Right: Having stunned the brute, you can now wander on through the myriad passages in your search for the Chalice.

```
PLATINUM SERPENT
IS STUNNED !
YOU HAVE USED  40 UNITS.
YOU HAVE 460 UNITS REMAINING.

TUNNELS LEAD TO
 15, 18, 21, 24
YOU ARE CURRENTLY IN CHAMBER 41
CREATE WALL, MOVE OR VIEW ?M
WHICH CHAMBER TO MOVE TO 44

TUNNELS LEAD TO
 15, 18, 21, 24
YOU ARE CURRENTLY IN CHAMBER 41
WHICH CHAMBER TO MOVE TO M

TUNNELS LEAD TO
 15, 18, 21, 24
YOU ARE CURRENTLY IN CHAMBER 41
WHICH CHAMBER TO MOVE TO 24

WARNING -
I CAN HEAR A SERPENT HISSING !

TUNNELS LEAD TO
  8, 41
YOU ARE CURRENTLY IN CHAMBER 24
CREATE WALL, MOVE OR VIEW ?_
```

VARIABLES

A$	The string holding the sequence of chambers.
B	Random flag for attack.
C	Chamber where the sorcerer and chalice are hidden.
F	General flag.
I	General loop count.
J	General loop count.
K	Stun flag.
L	Chamber where Magic Wall has been built.
M	Amount of magic left in Magic Staff.
N	Number of chambers.
P	General numeric reply.
Q$	General string reply.
S	Strength of an opposing force.
T	Temporary store.
U	Amount of magic used.
W	Flag to indicate win.
X	Contents of chamber.
Y	Player's chamber number.
Z	Chamber number for object.
Array AA	Contents of labyrinth.
Array BB	All possible connections from a chamber.

PROGRAM STRUCTURE

Statement	Function	Action
Lines 10-50	Set Up	Sets up arrays used in game. Prints title.
Lines 60-80	Call Subroutines	Invokes subroutines to print instructions, to load the map of the barrows and to zero the labyrinth array.
Lines 90-190	Positions	Calls subroutines to load the various characters into initial positions, also to give player initial position and to display this position.
Lines 200-250	Action	Player is given choice of three actions. Appropriate subroutine called to implement choice.
Lines 260-510	End Game	Logic for end of game and messages. Choice of replay.
Lines 520-670	Encounters	Arithmetic for Magic Staff. Messages and logic if character encountered.
Lines 680-1150	Move and Fight	Controls moves made by the player and handles logic when a fight occurs.
Lines 1160-1300	Wall	Creates Magic Wall when required and if circumstances permit.
Lines 1310-1460	View	Routine to 'view chamber'.
Lines 1470-1700	Positions	Routine to display position and give information about adjoining chambers.
Lines 1710-1730	Load Array	Loads the labyrinth array with a random set of 'troubles'.
Lines 1740-1770	Zero Array	Zeroes a number of arrays.
Lines 1780-1820	Map	Sets up 'map' of barrow chambers.
Lines 1830-1910	Instructions	Prints initial instructions.

This is not easy (the game wouldn't be much fun if it was) but there are three aids for your task: the sorcerer will not move to a chamber you have just left; he will not enter a chamber containing a goblin; and he will not enter a chamber in which you have created the Magic Wall.

This Magic Wall can be created at any point in the labyrinth but only one wall can be created at any one time. If you build a second wall, the first will fall down! Also, you cannot build a wall in a chamber which is already occupied — if you try, it will fall down staight away.

The final option, available when all else has failed (you've run out of magic or been killed), is the possibility of re-incarnation.

TECHNICAL DETAILS

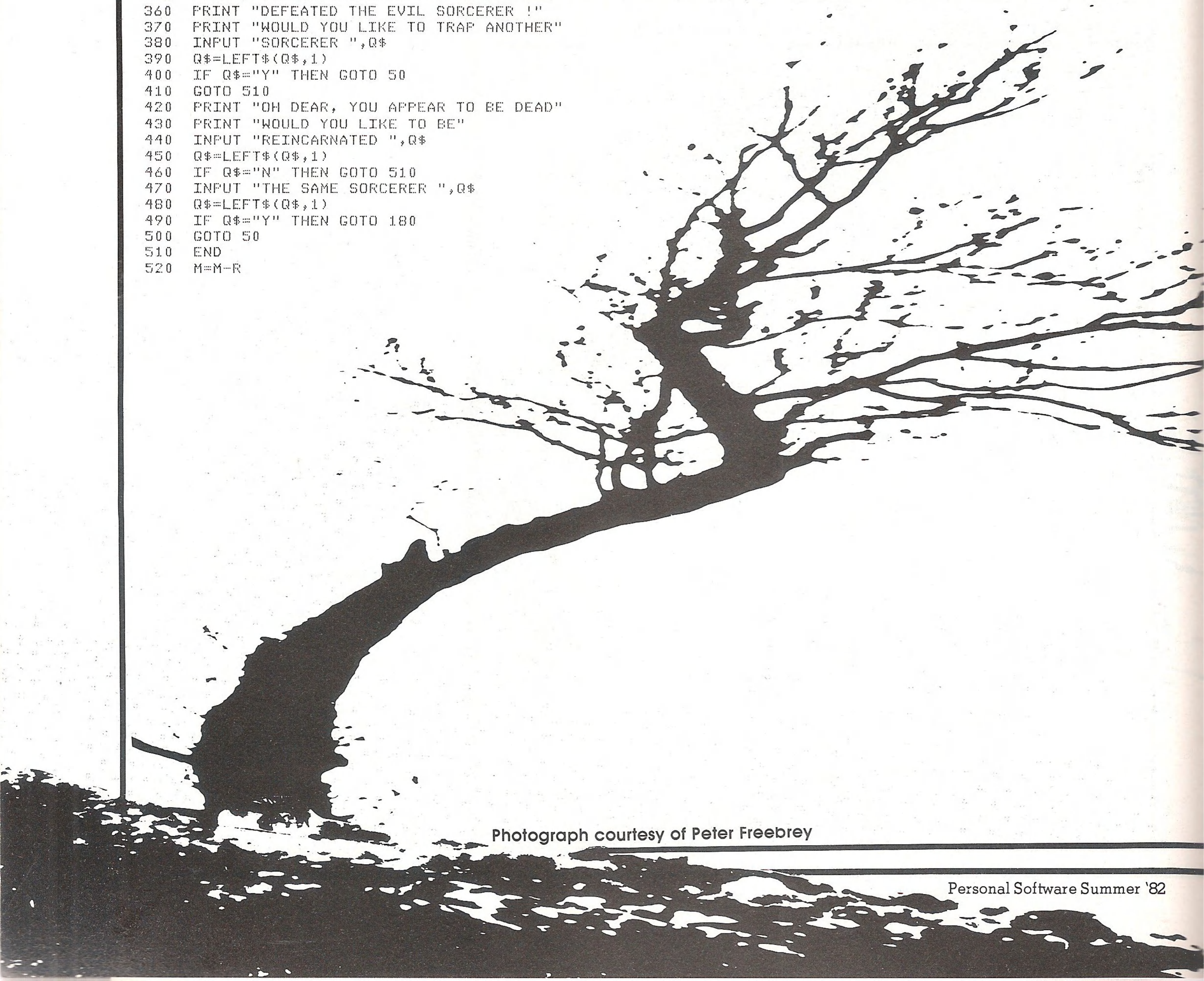
This is a conversion from ATOM BASIC. The only real difficulty encountered in the conversion was changing the ATOM's odd string handling into the more regular string functions BBC BASIC uses. The structure of the labyrinth is stored in A$ and is constructed from blocks of four characters containing the connections from the chamber whose number corresponds to the block; block 0 is a dummy. The chamber number is stored as an ASCII character whose code is the number plus 48. For example, chamber five connects to 1;<= which translates to chamber four connects to one, 11, 12 and 13.

HOW TO PLAY

The program includes very full instructions but you need to keep the copy of the map with you when you play. You also need either a very good memory or pen and paper so that you can keep a note of the position of the various 'nasties' you meet on your travels through the barrows — only the sorcerer actually moves spontaneously during the game, but, of course, you may alter the position of a dragon, serpent or dwarf by 'transporting' it.

▶

```
 10  REM**CONSULT TEXT FOR MAP
 20  B=0
 30  N=42:@1=&00000303
 40  DIM AA(N),BB(3)
 50  CLS:PRINT:PRINT "THE WHITE BARROWS"
 60  GOSUB 1830
 70  GOSUB 1780
 80  GOSUB 1740
 90  FOR X=1 TO 9
100  GOSUB 1710
110  NEXT X
120  FOR I=1 TO 3
130  X=10:GOSUB 1710
140  X=11:GOSUB 1710
150  NEXT I
160  X=0:GOSUB 1710:Y=R
170  GOSUB 1710:C=R
180  M=500:L=0
190  W=0:K=0:GOSUB 1470
200  INPUT "CREATE WALL, MOVE OR VIEW ",Q$
210  Q$=LEFT$(Q$,1)
220  IF Q$="V" THEN GOSUB 1310:GOTO 260
230  IF Q$="C" THEN GOSUB 1160:GOTO 260
240  IF Q$="M" THEN GOSUB 680:GOTO 260
250  GOTO 200
260  IF AA(Y)=0 OR K=1 OR AA(Y)=10 OR
     AA(Y)=11 THEN  GOTO 300
270  PRINT "THE ";:Z=Y:GOSUB 560
280  PRINT "SUDDENLY ATTACKED !":PRINT
290  GOTO 420
300  IF M>0 THEN GOTO 340
310  PRINT "YOUR STAFF IS NOW USELESS AND"
320  PRINT "THE EVIL FORCE ATTACKS !"
330  GOTO 420
340  IF W=0 THEN GOTO 190
350  PRINT "CONGRATULATIONS YOU HAVE"
360  PRINT "DEFEATED THE EVIL SORCERER !"
370  PRINT "WOULD YOU LIKE TO TRAP ANOTHER"
380  INPUT "SORCERER ",Q$
390  Q$=LEFT$(Q$,1)
400  IF Q$="Y" THEN GOTO 50
410  GOTO 510
420  PRINT "OH DEAR, YOU APPEAR TO BE DEAD"
430  PRINT "WOULD YOU LIKE TO BE"
440  INPUT "REINCARNATED ",Q$
450  Q$=LEFT$(Q$,1)
460  IF Q$="N" THEN GOTO 510
470  INPUT "THE SAME SORCERER ",Q$
480  Q$=LEFT$(Q$,1)
490  IF Q$="Y" THEN GOTO 180
500  GOTO 50
510  END
520  M=M-R
530  PRINT "YOU HAVE USED "R" UNITS."
540  PRINT "YOU HAVE "M" UNITS
     REMAINING.":PRINT
550  RETURN
560  ON AA(Z) GOTO 570,580,590,600,610,
     620,630,640,650,660,670
570  PRINT "RED DRAGON":S=30:RETURN
580  PRINT "GREEN DRAGON":S=25:RETURN
590  PRINT "YELLOW DRAGON":S=20:RETURN
600  PRINT "PLATINUM SERPENT":S=28:RETURN
610  PRINT "GOLD SERPENT":S=23:RETURN
620  PRINT "SILVER SERPENT":S=18:RETURN
630  PRINT "DWARF WITH A SABRE":S=24:RETURN
640  PRINT "DWARF WITH A SWORD":S=19:RETURN
650  PRINT "DWARF WITH A KNIFE":S=14:RETURN
660  PRINT "FRIENDLY GNOME":RETURN
670  PRINT "MAGIC SPOT"':RETURN
680  INPUT "WHICH CHAMBER TO MOVE TO "P
690  F=0:PRINT
700  FOR I=0 TO 3
710  IF BB(I)=P THEN F=1
720  NEXT I
730  IF P=0 OR F=0 THEN GOSUB 1470:GOTO 680
740  IF C<>P THEN GOTO 860
750  W=1:F=0
760  FOR I=0 TO 3
770  T=ASC(MID$(A$,C*4-4+I+1,1))-48
780  IF T=0 OR T=Y OR AA(T)=10 OR T=L
     THEN GOTO 810
790  IF W=1 OR RND(1)<.5 THEN C=T
800  W=0:IF AA(T)=11 THEN F=1
810  NEXT I
820  IF W=1 THEN RETURN
830  IF F=0 THEN PRINT "THE SORCERER
     HAS RUN OFF !":GOTO 860
```



Photograph courtesy of Peter Freebrey

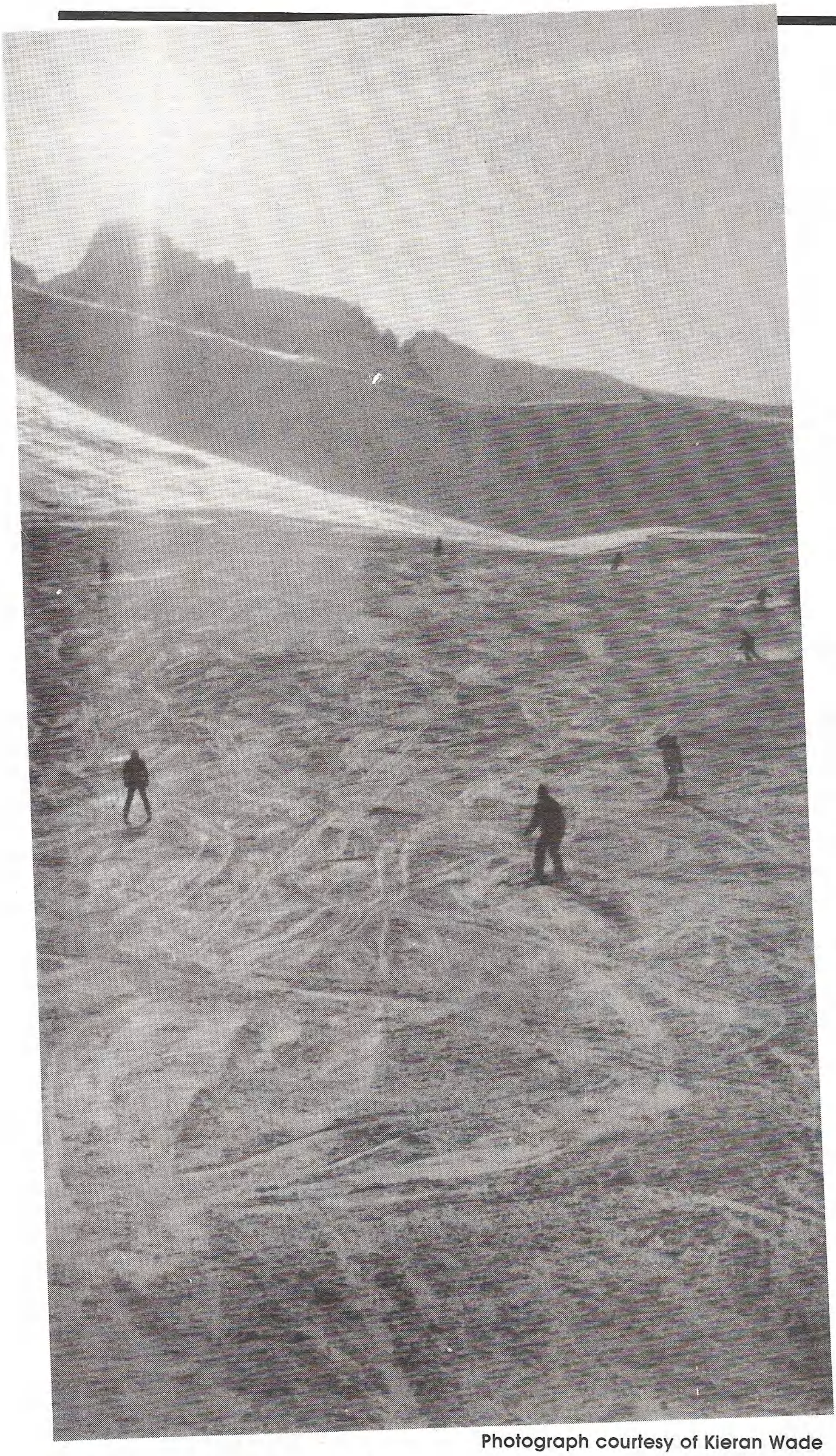
```
840  PRINT "THE SORCERER TROD ON THE MAGIC"
850  PRINT "SPOT !":X=0:GOSUB 1710:C=R:
     GOTO 740
860  Y=P:IF AA(P)=0 THEN RETURN
870  IF AA(P)<>11 THEN 910
880  PRINT "POWWW! YOU HAVE JUST TRODDEN ON"
890  PRINT "THE MAGIC SPOT!":PRINT
900  X=0:GOSUB 1710:Y=R:RETURN
910  IF AA(P)<>10 THEN GOTO 990
920  R=RND(20)+19
930  PRINT "THE GOBLIN WILL TELL YOU WHERE"
940  PRINT "THE SORCER IS FOR ";R;" GOLD"
950  INPUT "COINS. AGREED ",Q$
960  Q$=LEFT$(Q$,1)
970  IF Q$<>"Y" THEN RETURN
980  PRINT "THE SORCERER IS IN CHAMBER
     ";C;".":M=M-R:GOTO 1140
990  K=0:Z=Y:GOSUB 560:PRINT "IS ABOUT
     TO ATTACK !"
1000 PRINT "YOU HAVE THREE CHOICES:":PRINT
1010 PRINT "1 - STUN IT."
1020 PRINT "2 - TRANSPORT IT."
1030 PRINT "3 - DESTROY IT.":PRINT
1040 INPUT "YOUR CHOICE ",P
1050 IF P<1 OR P>3 THEN GOTO 1040
1060 INPUT "AMOUNT OF MAGIC ",R
1070 IF R<0 OR R>M THEN GOTO 1060
1080 CLS:GOSUB 560:PRINT "IS ";:B=0
1090 IF P=1 AND R>S+(RND(2*S/3)-S/3-1)
     THEN PRINT "STUNNED !":K=1
1100 IF P<>2 OR R<S*2+RND(2*S)-S THEN
     GOTO 1120
1110 PRINT "TRANSPORTED":X=AA(Y):
     AA(Y)=0:GOSUB 1710
1120 IF P=3 AND R>S*4+RND(2*S)-S THEN
     PRINT "DESTROYED !"':AA(Y)=0
1130 IF K=0 AND AA(Y)<>0 THEN PRINT
     "DAZED !":B=RND(8)-1
1140 GOSUB 520:IF B>2 AND M>0 THEN GOTO 1000
1150 RETURN
1160 PRINT "WHICH CHAMBER DO YOU WISH TO"
1170 INPUT "CREATE THE MAGIC WALL IN ",L
1180 PRINT
1190 IF L<1 OR L>N THEN GOSUB 1470:GOTO 1160
1200 IF AA(L)<>0 THEN GOTO 1240
```

```
1210 PRINT "THE MAGIC WALL NOW EXISTS IN"
1220 PRINT "CHAMBER ";L
1230 GOTO 1270
1240 PRINT "ZAAPPPP ! CHAMBER ";L;" ALREADY":
     L=0
1250 PRINT "CONTAINS MAGIC, THE WALL HAS"
1260 PRINT "FAILED !"
1270 PRINT
1280 R=RND(20)-1+20
1290 GOSUB 520
1300 RETURN
1310 PRINT "WHICH CHAMBER DO YOU WISH TO"
1320 INPUT "VIEW ",P
1330 F=0
1340 FOR I=0 TO 3
1350 IF P=BB(I) THEN F=1
1360 NEXT I
1370 IF P=0 OR F=0 THEN PRINT "CHAMBER
     MUST BE CONNECTED TO YOURS":GOTO 1310
1380 PRINT
1390 IF P=C THEN PRINT "YOU HAVE FOUND
     THE SORCERER !"
1400 IF AA(P)=0 THEN PRINT "THE
     CHAMBER IS EMPTY !":GOTO 1440
1410 PRINT "THERE IS A ";
1420 Z=P:GOSUB 560
1430 PRINT "IN CHAMBER ";P;".";
1440 R=RND(20)+9
1450 GOSUB 520
1460 RETURN
1470 F=0:Q$="1234"
1480 FOR I=0 TO 3
1490 BB(I)=ASC(MID$(A$,Y*4-4+I+1,1))-48
1500 F=F+AA(BB(I))-(100*ABS(BB(I)=C))
1510 NEXT I
1520 IF F=0 THEN GOTO 1640
1530 PRINT "WARNING -"
1540 IF F<0 THEN PRINT "THE CHALICE IS NEAR !"
1550 FOR I=0 TO 3
1560 R=RND(4):IF MID$(Q$,R,1)=" " THEN
     GOTO 1560
1570 Q$=LEFT$(Q$,R-1)+" "+RIGHT$(Q$,4-R):
     T=AA(BB(R-1))
1580 IF T>0 AND T<4 THEN PRINT "I CAN
     HEAR A DRAGON !"
1590 IF T>3 AND T<7 THEN PRINT "I CAN
     HEAR A SERPENT HISSING !"
1600 IF T>6 AND T<11 THEN PRINT "I CAN
     HEAR  MONEY RATTLING !"
1610 IF T=11 THEN PRINT "I CAN FEEL
     MAGIC  VERY CLOSE !"
1620 NEXT I
1630 PRINT
1640 PRINT "TUNNELS LEAD TO"
1650 FOR I=0 TO 3
1660 IF BB(I)>0 THEN PRINT BB(I);",";
1670 NEXT I
1680 PRINT CHR$(8);" "
1690 PRINT "YOU ARE CURRENTLY IN CHAMBER ";Y
1700 RETURN
1710 R=RND(42):IF AA(R)>0 GOTO 1710
1720 AA(R)=X
1730 RETURN
1740 FOR I=0 TO 42
1750 AA(I)=0
1760 NEXT I
1770 RETURN
1780 A$="23401567189:
     1;<=2>?@2ABC2DEF3GHI3JKL3MNO4PQR4STU"
1790 A$=A$+"4VWX5@X05Y005>A06@C06Y006AD0
     7CF07Y007DG08FI08Y00"
1800 A$=A$+"8GJ09IK09JL09KM0:L00:Z00:
     MP0;OR0;Z00;PS0<RU0<Z00"
1810 A$=A$+"<SV0=UX0=Z00=>V0?BEHNQTW"
1820 RETURN
1830 PRINT:PRINT"YOUR MISSION IS TO
     TRAP THE"
1840 PRINT "EVIL SORCERER AND RETURN THE"
1850 PRINT "MAGIC CHALICE."
1860 PRINT "YOU HAVE BEEN GIVEN A MAGIC"
1870 PRINT "STAFF WITH 500 UNITS OF MAGIC."
1880 PRINT "USE IT TO OVERCOME THE"
1890 PRINT "HAZARDS HIDDEN IN THE
     WHITE"
1900 INPUT "BARROWS.  HIT RETURN TO
     BEGIN.",Q$:CLS
1910 RETURN
```

Converted from a program by R L Tucker

SKI~RUN

Practice your ski-ing techniques on this fast-moving slalom run. (Another case of schussing on the piste?)



Photograph courtesy of Kieran Wade

This program simulates a slalom ski run, complete with gates you have to negotiate. The object of the game is simple, you must reach the bottom of the course in the shortest possible time without missing any gates and without going off the edge of the course.

There are two kinds of course available: a standard pre-programmed run of quite reasonable difficulty and a randomly generated course, which is usually easier. In both cases you can preview your course.

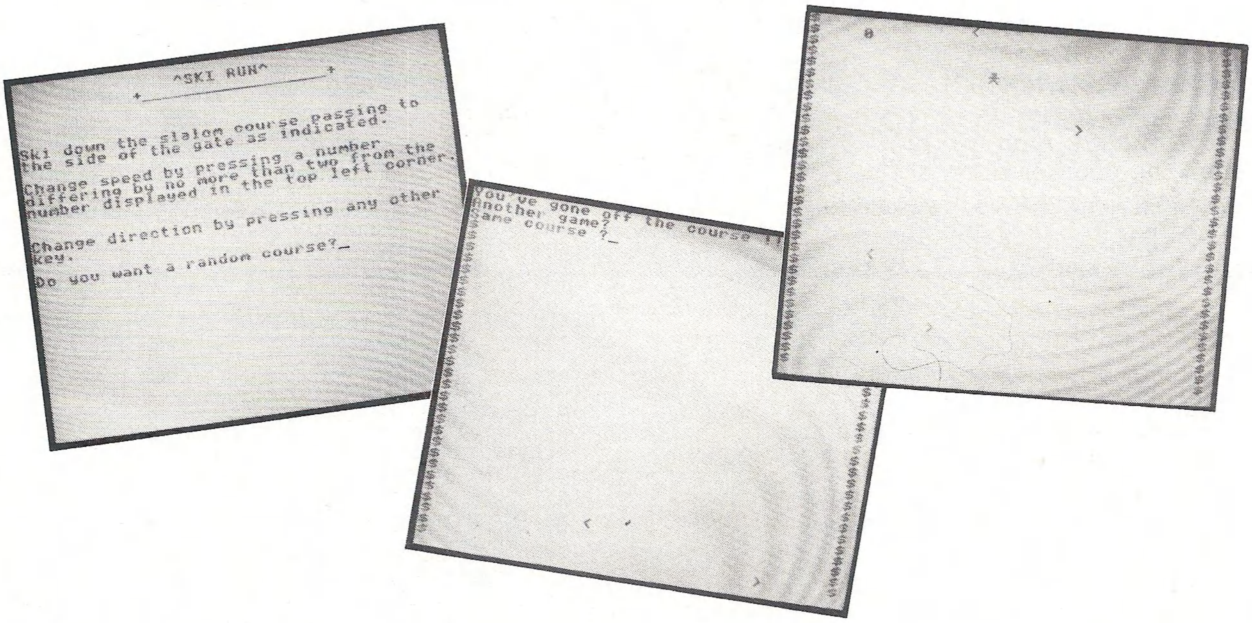
As you gain experience, you may wish to ski faster. The program permits this very easily — the game may be speeded up by inputting a number not greater than two digits larger than the number displayed in the top-left hand corner (it starts each game at zero).

TECHNICAL DETAILS

Ski-Run was converted from a Sharp MZ80 program. The most interesting point of the conversion is the section which creates the 'little man' graphics character. This is done in the usual way with a VDU 23 command. It is very tempting to try to create an over complex graphics character. This should be avoided however, as the minimum width for a clear feature in a graphics character is two dots. Still, if you want to experiment, try adding some skis to the figure!

HOW TO PLAY

The computer displays instructions and asks you to choose a random course or the set course. You are given the option of viewing the course before you start. When the game starts it is actually the



```
^SKI RUN^
*__________________*

Ski down the slalom course passing to
the side of the gate as indicated.

Change speed by pressing a number
differing by no more than two from the
number displayed in the top left corner.

Change direction by pressing any other
key.
Do you want a random course?_
```

```
You've gone off the course
Another game?
Same course ?_
```

PROGRAM STRUCTURE

Statement	Function	Action
Lines 10-30	Set Up	Selects MODE 4, defines character of 'little man', sets background colour to white and foreground colour to blue.
Lines 40-170	Text	Displays text at beginning of game.
Lines 180-240	Select Course	Gives player the option of a random rather than the pre-set course.
Lines 250-280	Set Course	Sets up the 'gates' on the chosen course.
Lines 290-320	Preview	Gives player the option of viewing the course before attempting it.
Lines 330-540	Print/Move Course	Prints the course and moves it up the screen.
Lines 550-690	Move Man	Displays start message and figure of the man, controls change of direction when any key pressed and checks to see whether man is off the course.
Lines 700-770	Gate	Checks to see whether man passes to correct side of gate — if not, adds 5 to penalty score and sounds a 'bleep'.
Lines 780-870	Random Course	Works out and displays a random course.
Lines 880-950	End Game	If man goes off the course, message printed to confirm this. If course completed, prints time taken, penalty time and the sum of the two.
Lines 960-1050	Restart	Gives option of another game with the same or a different course.

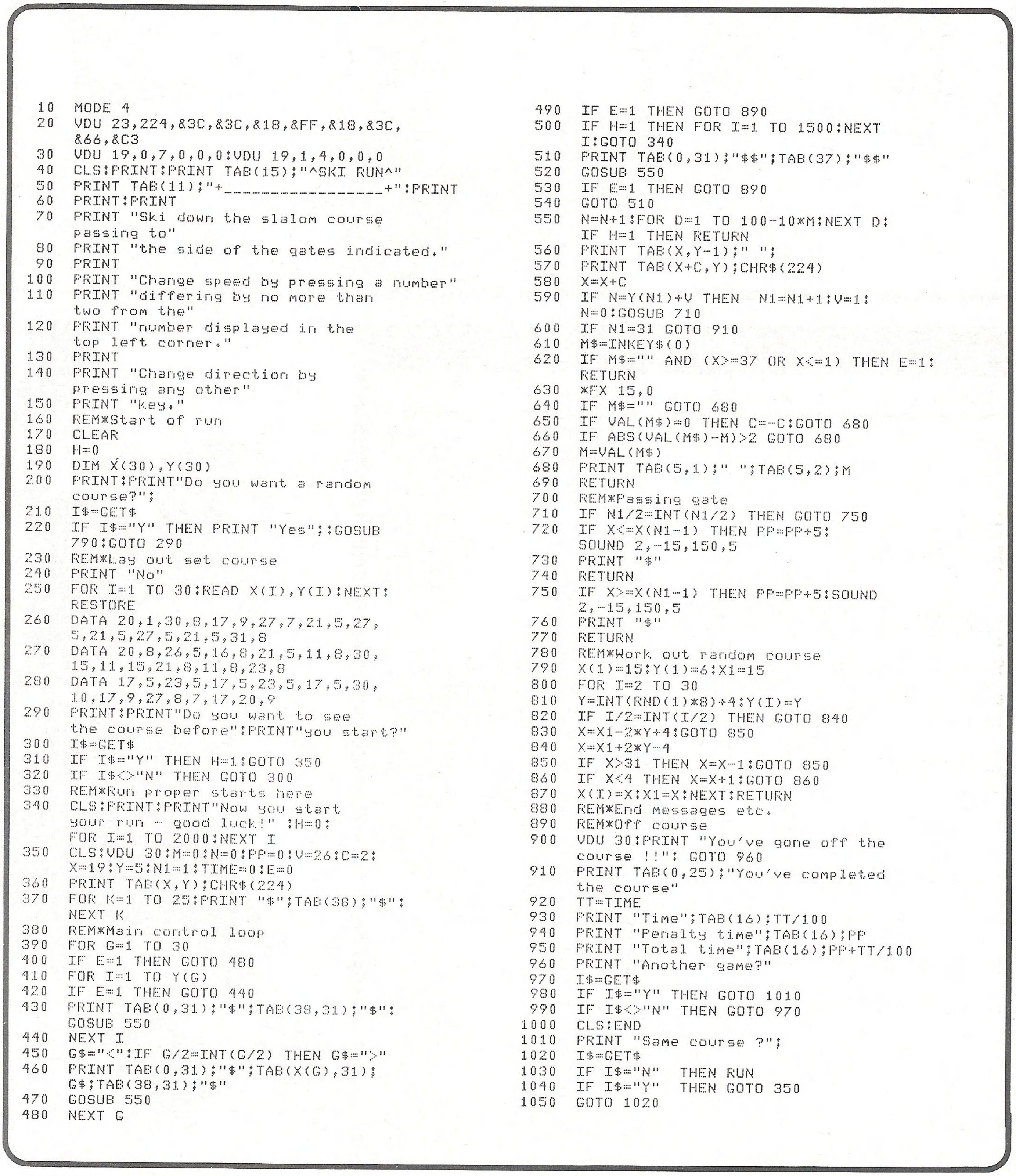
▶

course that moves up the screen. All the skier can do is to change direction — which is done by hitting any key. If you wish to speed up the game you can increase the number displayed at the top of the screen ***by no more than two at any time.*** The counter starts at 0 — so you can alter it to 2 if you wish, and then to 3 or 4 and so on. It will always revert to 0 at the beginning of every game.

You have to go around the gates on the course on the side indicated by the arrow. If you go round them the wrong way a 'bleep' will sound and you'll be awarded five penalty points; if you crash into the barrier you have to start again.

When you complete the course successfully the computer displays your 'time' (the actual time taken), your 'penalty time' (the number of penalty points you obtained) and your 'total time' (the sum of the two).

Each time you restart the game you can choose to try the same course or a new one. Happy skiing!



```
  10  MODE 4
  20  VDU 23,224,&3C,&3C,&18,&FF,&18,&3C,
      &66,&C3
  30  VDU 19,0,7,0,0,0:VDU 19,1,4,0,0,0
  40  CLS:PRINT:PRINT TAB(15);"^SKI RUN^"
  50  PRINT TAB(11);"+_________________+":PRINT
  60  PRINT:PRINT
  70  PRINT "Ski down the slalom course
      passing to"
  80  PRINT "the side of the gates indicated."
  90  PRINT
 100  PRINT "Change speed by pressing a number"
 110  PRINT "differing by no more than
      two from the"
 120  PRINT "number displayed in the
      top left corner."
 130  PRINT
 140  PRINT "Change direction by
      pressing any other"
 150  PRINT "key."
 160  REM*Start of run
 170  CLEAR
 180  H=0
 190  DIM X(30),Y(30)
 200  PRINT:PRINT"Do you want a random
      course?";
 210  I$=GET$
 220  IF I$="Y" THEN PRINT "Yes";:GOSUB
      790:GOTO 290
 230  REM*Lay out set course
 240  PRINT "No"
 250  FOR I=1 TO 30:READ X(I),Y(I):NEXT:
      RESTORE
 260  DATA 20,1,30,8,17,9,27,7,21,5,27,
      5,21,5,27,5,21,5,31,8
 270  DATA 20,8,26,5,16,8,21,5,11,8,30,
      15,11,15,21,8,11,8,23,8
 280  DATA 17,5,23,5,17,5,23,5,17,5,30,
      10,17,9,27,8,7,17,20,9
 290  PRINT:PRINT"Do you want to see
      the course before":PRINT"you start?"
 300  I$=GET$
 310  IF I$="Y" THEN H=1:GOTO 350
 320  IF I$<>"N" THEN GOTO 300
 330  REM*Run proper starts here
 340  CLS:PRINT:PRINT"Now you start
      your run - good luck!" :H=0:
      FOR I=1 TO 2000:NEXT I
 350  CLS:VDU 30:M=0:N=0:PP=0:V=26:C=2:
      X=19:Y=5:N1=1:TIME=0:E=0
 360  PRINT TAB(X,Y);CHR$(224)
 370  FOR K=1 TO 25:PRINT "$";TAB(38);"$":
      NEXT K
 380  REM*Main control loop
 390  FOR G=1 TO 30
 400  IF E=1 THEN GOTO 480
 410  FOR I=1 TO Y(G)
 420  IF E=1 THEN GOTO 440
 430  PRINT TAB(0,31);"$";TAB(38,31);"$":
      GOSUB 550
 440  NEXT I
 450  G$="<":IF G/2=INT(G/2) THEN G$=">"
 460  PRINT TAB(0,31);"$";TAB(X(G),31);
      G$;TAB(38,31);"$"
 470  GOSUB 550
 480  NEXT G
 490  IF E=1 THEN GOTO 890
 500  IF H=1 THEN FOR I=1 TO 1500:NEXT
      I:GOTO 340
 510  PRINT TAB(0,31);"$$";TAB(37);"$$"
 520  GOSUB 550
 530  IF E=1 THEN GOTO 890
 540  GOTO 510
 550  N=N+1:FOR D=1 TO 100-10*M:NEXT D:
      IF H=1 THEN RETURN
 560  PRINT TAB(X,Y-1);" ";
 570  PRINT TAB(X+C,Y);CHR$(224)
 580  X=X+C
 590  IF N=Y(N1)+V THEN  N1=N1+1:V=1:
      N=0:GOSUB 710
 600  IF N1=31 GOTO 910
 610  M$=INKEY$(0)
 620  IF M$="" AND (X>=37 OR X<=1) THEN E=1:
      RETURN
 630  *FX 15,0
 640  IF M$="" GOTO 680
 650  IF VAL(M$)=0 THEN C=-C:GOTO 680
 660  IF ABS(VAL(M$)-M)>2 GOTO 680
 670  M=VAL(M$)
 680  PRINT TAB(5,1);" ";TAB(5,2);M
 690  RETURN
 700  REM*Passing gate
 710  IF N1/2=INT(N1/2) THEN GOTO 750
 720  IF X<=X(N1-1) THEN PP=PP+5:
      SOUND 2,-15,150,5
 730  PRINT "$"
 740  RETURN
 750  IF X>=X(N1-1) THEN PP=PP+5:SOUND
      2,-15,150,5
 760  PRINT "$"
 770  RETURN
 780  REM*Work out random course
 790  X(1)=15:Y(1)=6:X1=15
 800  FOR I=2 TO 30
 810  Y=INT(RND(1)*8)+4:Y(I)=Y
 820  IF I/2=INT(I/2) THEN GOTO 840
 830  X=X1-2*Y+4:GOTO 850
 840  X=X1+2*Y-4
 850  IF X>31 THEN X=X-1:GOTO 850
 860  IF X<4 THEN X=X+1:GOTO 860
 870  X(I)=X:X1=X:NEXT:RETURN
 880  REM*End messages etc.
 890  REM*Off course
 900  VDU 30:PRINT "You've gone off the
      course !!": GOTO 960
 910  PRINT TAB(0,25);"You've completed
      the course"
 920  TT=TIME
 930  PRINT "Time";TAB(16);TT/100
 940  PRINT "Penalty time";TAB(16);PP
 950  PRINT "Total time";TAB(16);PP+TT/100
 960  PRINT "Another game?"
 970  I$=GET$
 980  IF I$="Y" THEN GOTO 1010
 990  IF I$<>"N" THEN GOTO 970
1000  CLS:END
1010  PRINT "Same course ?";
1020  I$=GET$
1030  IF I$="N"  THEN RUN
1040  IF I$="Y"  THEN GOTO 350
1050  GOTO 1020
```

Based on an idea by Henry Budgett and Dave Sinfield.

ALIEN ATTACKERS

Your Planet is under attack and you are in sole control.

EDWARD L. ALPERSON
presents
INVADERS FROM M
Photographed in
COLOR

Photograph courtesy of 20th Century Fox

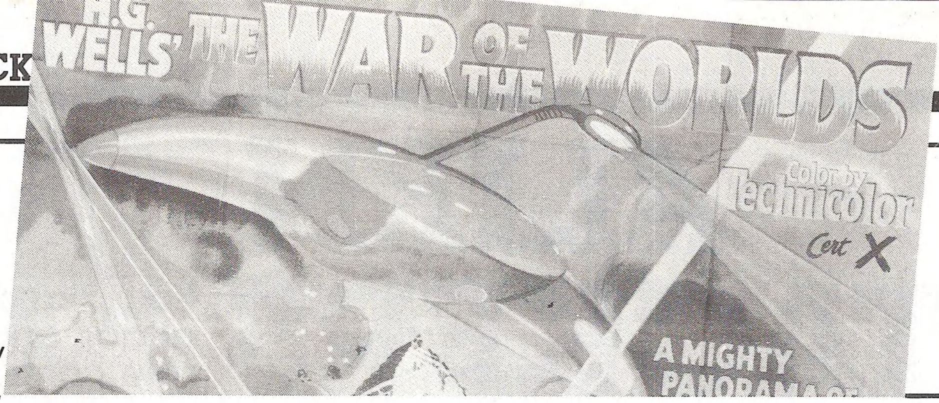
It is the 25th century. Earth has been abandoned after the onset of a world-wide ice age which began in the northern hemisphere in the 21st Century and gradually engulfed the entire planet. The peoples of earth dispersed over far-flung galaxies but their emnities survived and each colony is constantly under threat of attack and in a state of preparedness, both for defence and aggression.

On this occasion, however, the enemy has duped your early warning system. Your fleet of battle craft is out on its own war-mongering mission and you remain as the only person on the planet able to operate a space ship and launch the deadly laser bolts — the only weapons which have any effect on the enemy craft.

The Aliens Are Coming

To make matters worse *all* your sophisticated warships are away from the base — there is only a single tiny training ship left. It can only fire one laser bolt at a time and can only hover above the oncoming craft, moving to the left and right but unable to advance or retreat.

The enemy's twenty battle cruisers glide silently towards you. You know that they will not attempt to fire at your fast-moving craft as it is so small their crude weapons could not locate it accurately but, if any more than five of them land, your planet's ground-based forces will be overwhelmed. So you are faced with the challenge of destroying three-quarters of the entire enemy fleet single-handed with these minimal resources — or else your colony will suffer devastating losses. ▶



Photograph courtesy of Paramount

PROGRAM STRUCTURE

Statement	Function	Action
Lines 10-40	Set Up	Selects MODE 4, calls routines to initialise variables and to print title and instructions. Selects MODE 4 a second time and resets scrolling mode.
Lines 50-80	Run Game	Calls routines to play game and finish it.
Lines 90-240	DEFPROCINIT	Defines graphics characters and initialises variables.
Lines 250-400	DEFPROCTEXT	Displays text at the start of play.
Lines 410-520	DEFPROCPLAY	Initialises variables for play, calls routine to display the ship, calls routine to change direction of the ship, to fire bolts, to display the aliens and to move them up the screen.
Lines 530-610	DEFPROCSHIP	Moves the ship, calls routine to crash the ship when appropriate and displays the ship.
Lines 620-690	DEFPROCDIR	Changes direction of ship when 'R' is pressed.
Lines 700-770	DEFPROCALIEN	Calls scrolling routine, counts aliens and displays them.
Lines 780-840	DEFPROC-SCROLL	Moves whole screen in order to produce scrolling effect.
Lines 850-930	DEFPROCBOLT	Prints and moves the bolt and emits sound as the bolt travels. Logic to detect a dead hit and a near miss.
Lines 940-1050	DEFPROCBOOM	Destroys alien and emits sound of explosion.
Lines 1060-1140	DEFPROCNEAR	Signals a near miss by appropriate sound and a flash.
Lines 1150-1180	DEFPROCDELAY	Delay loop.
Lines 1190-1250	DEFPROCZAP	Destroys ship, with sound effect, if it collides with an alien.
Lines 1260-1380	DEFPROCFIN	Prints 'win' or 'lose' message at the end of your turn and offers the choice of another game.
Lines 1390-1420	Screen Address	Converts x,y co-ordinates to memory mapped screen address.

TECHNICAL DETAILS

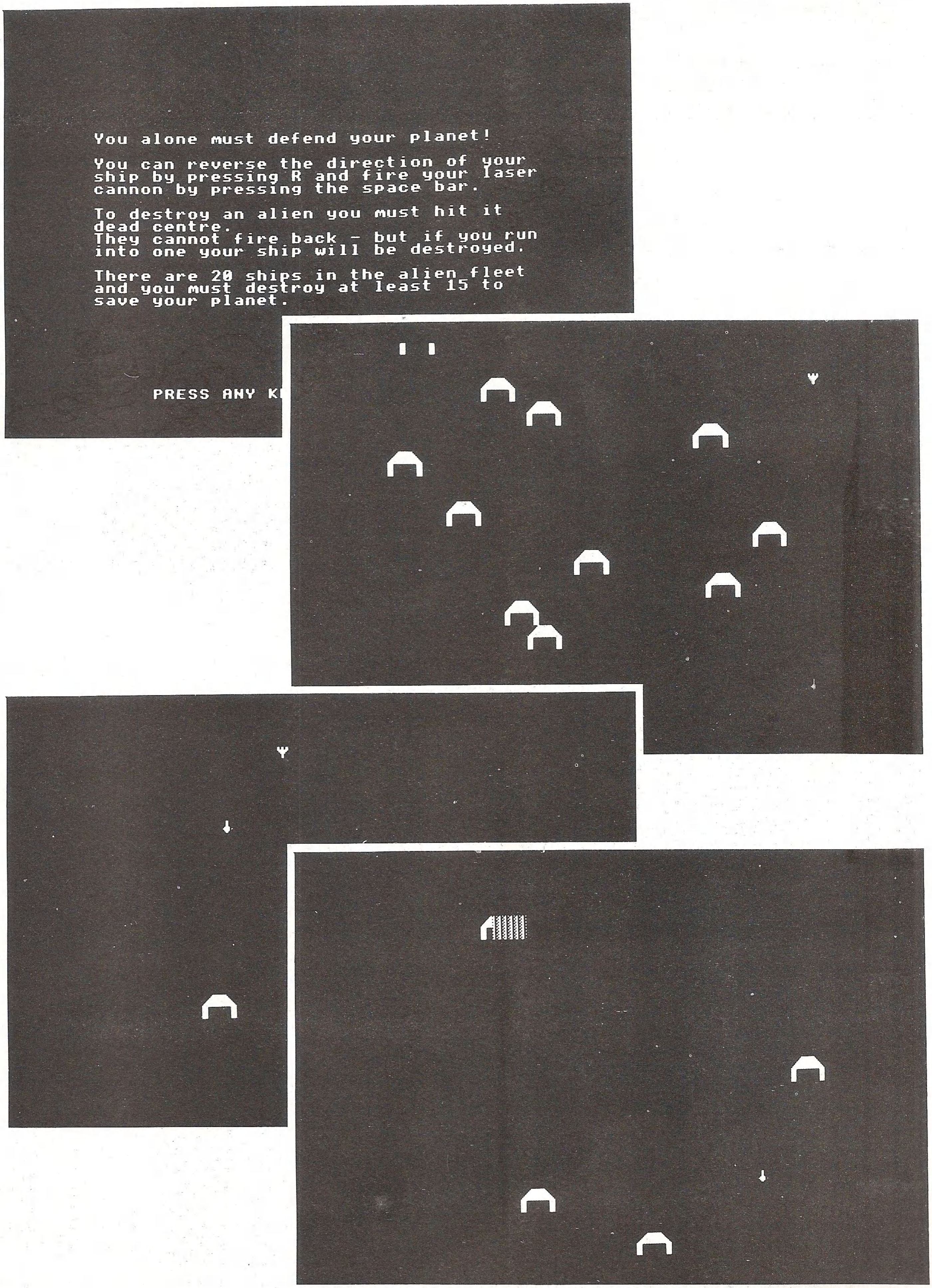
Alien Attack uses character graphics in much the same way that Fox and Hounds, Breakthough and Squish do. The procedure INIT sets up the characters required, including CHR$(230) the small space ship. The alien ships are moved up the screen by scrolling the entire display. The only complication is that the BBC Micro performs a scroll by changing the memory address of each line. This only causes a problem because of the need to PEEK screen locations to find out what is already present. The usual FNS function has to be modified to include a count of the number of times the screen has been scrolled — the variable SC.

Notice also the use of the noise channel zero in the procedures which destroy the aliens and the defender.

HOW TO PLAY

You have only one ship and it can only move to the left and right in a fixed band at the top of the screen. You can change the direction the ship is travelling in by pressing 'R'. To fire your laser cannon, hit the space bar; your cannon will not re-load until the previous bolt has been finished with — either by hitting an alien or by going off the bottom of the screen.

To destroy an alien battle cruiser you have to hit it dead centre. If you hit it on either flank a 'near miss' will register — but it will not help you in your mission to allow no more than five enemy craft to pass by you and land on your planet. Be careful though, if you collide into one of the enemy you will be instantly destroyed.



```
You alone must defend your planet!

You can reverse the direction of your
ship by pressing R and fire your laser
cannon by pressing the space bar.

To destroy an alien you must hit it
dead centre.
They cannot fire back - but if you run
into one your ship will be destroyed.

There are 20 ships in the alien fleet
and you must destroy at least 15 to
save your planet.


     PRESS ANY K
```

▶

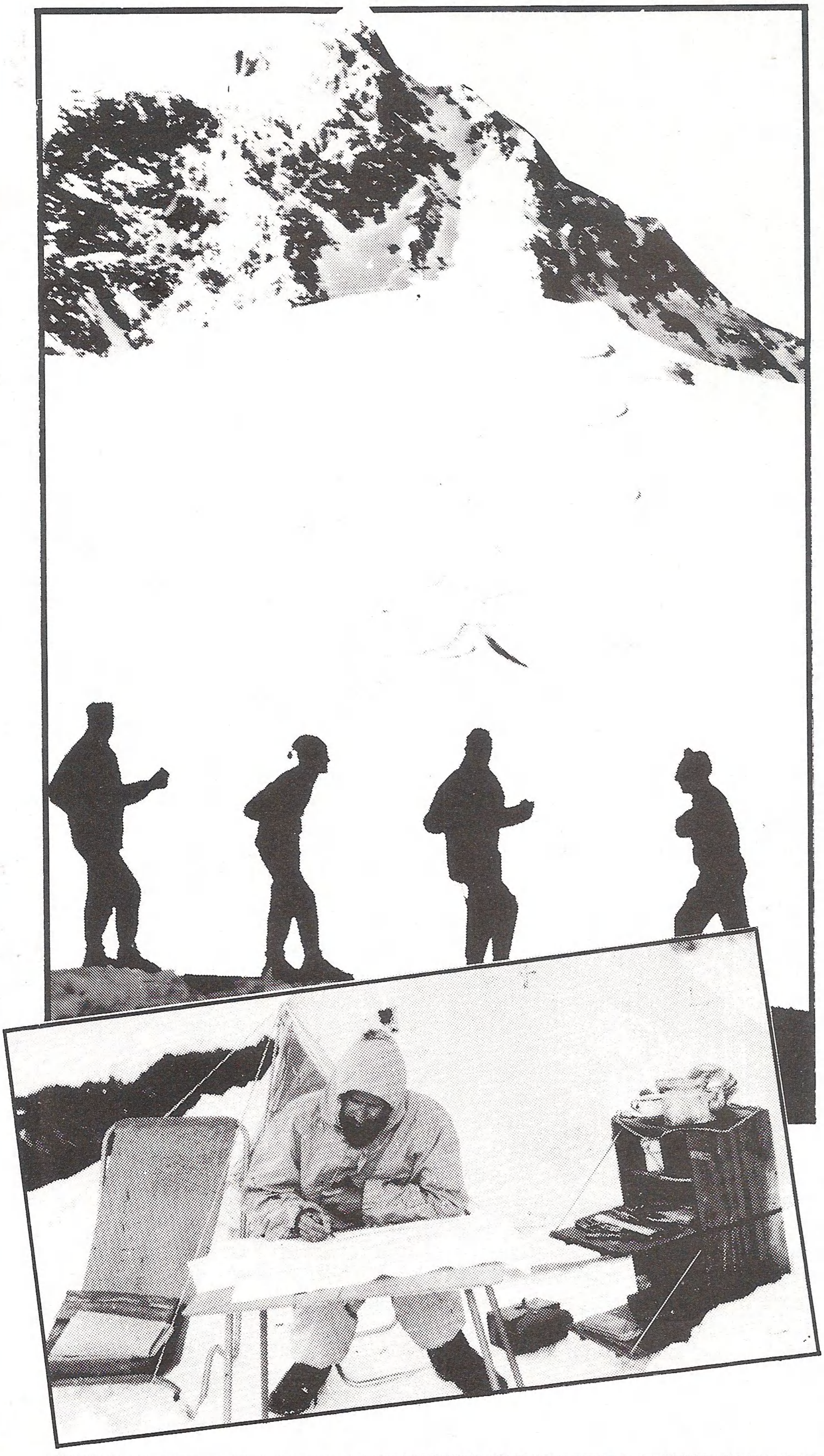
```
  10  MODE 4
  20  PROCINIT
  30  PROCTEXT
  40  MODE 4
  50  PROCPLAY
  60  CLS
  70  PROCFIN
  80  END
  90  DEFPROCINIT
 100  VDU23,224,&FF,&FF,&FF,&FF,&FF,&FF,&FF,&FF
 110  VDU23,225,&DB,&DB,&FF,&7E,&3C,&18,&18,&18
 120  VDU23,226,&01,&03,&07,&0F,&1F,&3F,&75,&FF
 130  VDU23,227,&80,&C0,&E0,&F0,&F8,&FC,&FE,&FF
 140  VDU23,228,&F0,&F0,&F0,&F0,&F0,&F0,&F0,&F0
 150  VDU23,229,&0F,&0F,&0F,&0F,&0F,&0F,&0F,&0F
 160  VDU23,230,&18,&18,&18,&18,&3C,&7D,&3C,&18
 170  VDU23,231,&55,&CC,&55,&CC,&55,&CC,&55,&CC
 180  SC=32
 190  G=0
 200  E=0
 210  S=20
 220  D=0
 230  EF=0
 240  ENDPROC
 250  DEFPROCTEXT
 260  CLS
 270  PRINT TAB(8,3);"A L I E N
      A T T A C K"
 280  PRINT TAB(0,9);"You alone must
      defend your planet!"
 290  PRINT:PRINT"You can reverse the
      direction of your"
 300  PRINT"ship by pressing R and fire
      your laser"
 310  PRINT"cannon by pressing the space bar."
 320  PRINT:PRINT"To destroy an alien
      you must hit it"
 330  PRINT"dead centre."
 340  PRINT"They cannot fire back - but
      if you run"
 350  PRINT"into one your ship will be
      destroyed."
 360  PRINT:PRINT"There are 20 ships in
      the alien fleet"
 370  PRINT"and you must destroy at least 15
      to"
 380  PRINT"save your planet."
 390  PRINT TAB(5,30);"PRESS ANY KEY TO
      BEGIN":A$=GET$
 400  ENDPROC
 410  DEFPROCPLAY
 420  X=10:Y=3:A=1:FF=0
 430  XA=20:YA=29
 440  PROCSHIP
 450  IF E=1 OR EF>26 THEN ENDPROC
 460  PROCDIR
 470  IF F=1 AND FF=0 THEN XB=X:YB=Y:FF=1
 480  IF FF=1 THEN PROCBOLT ELSE PROCDELAY
 490  IF RND(1)<.05 AND FF=0 THEN
      XA=RND(35)+2: PROCALIEN
 500  IF G=0 THEN PROCALIEN
 510  IF RND(1)<.04  THEN PROCSCROLL
 520  GOTO 440
 530  DEFPROCSHIP
 540  X=X+A
 550  IF X<1 THEN X=X+1
 560  IF X>39 THEN X=X-1
 570  M=?FNS(X,Y,0)
 580  IF M=&FF OR M=&01 OR M=&80 OR
      M=&F0 OR M=&0F THEN E=1:PROCZAP:ENDPROC
 590  IF A=1 THEN PRINT TAB(X-1,Y);" ";
      TAB(X,Y);CHR$(225);ELSE PRINT TAB(X+1,Y);
      " ";TAB(X,Y);CHR$(225);
 600  PRINT TAB(0,0);
 610  ENDPROC
 620  DEFPROCDIR
 630  F=0
 640  A$=INKEY$(0)
 650  *FX 15,0
 660  IF A$="" THEN ENDPROC
 670  IF A$="R" THEN A=-A
 680  IF A$=" " THEN F=1
```

```
 690  ENDPROC
 700  DEFPROCALIEN
 710  PROCSCROLL
 720  IF EF>0 THEN ENDPROC
 730  IF G=S THEN EF=1:S=0:ENDPROC
 740  G=G+1
 750  PRINT TAB(XA,YA);CHR$(226);
      CHR$(224);CHR$(227);
 760  PRINT TAB(XA,YA+1);CHR$(228);" ";
      CHR$(229);
 770  ENDPROC
 780  DEFPROCSCROLL
 790  SC=SC+2
 800  IF EF>0 THEN EF=EF+2
 810  PRINT TAB(X,Y);" ";
 820  IF FF=1 THEN PRINT TAB(XB,YB);" ";
 830  PRINT TAB(0,31):PRINT
 840  ENDPROC
 850  DEFPROCBOLT
 860  YB=YB+1
 870  PRINT TAB(XB,YB-1);" ";
 880  IF YB>29 THEN FF=0:ENDPROC
 890  SOUND 2,-15,YB+50,10
 900  PRINT TAB(XB,YB);CHR$(230);TAB(0,0);
 910  IF ?FNS(XB,YB+1,0)=&FF THEN D=D+1:
      PROCBOOM
 920  IF ?FNS(XB,YB+1,7)=&FF THEN PROCNEAR
 930  ENDPROC
 940  DEFPROCBOOM
 950  SOUND 0,-15,14,20
 960  PRINT TAB(XB,YB);" ";
 970  FOR I=1 TO 3
 980  PRINT TAB(XB-1,YB+1);STRING$(3,CHR$(231))
 990  PRINT TAB(XB-1,YB+2);STRING$(3,CHR$(231))
1000  FOR J=1 TO 100:NEXT J
1010  PRINT TAB(XB-1,YB+1);SPC(3);
1020  PRINT TAB(XB-1,YB+2);SPC(3);
1030  NEXT
1040  FF=0
1050  ENDPROC
1060  DEFPROCNEAR
1070  FF=0
1080  SOUND 0,-15,12,15
1090  FOR I=1 TO 10
1100  PRINT TAB(XB,YB);CHR$(231);
1110  PRINT TAB(XB,YB);" ";
1120  NEXT I
1130  PRINT TAB(XB,YB);" ";
1140  ENDPROC
1150  DEFPROCDELAY
1160  FOR D1=1 TO 50
1170  NEXT D1
1180  ENDPROC
1190  DEFPROCZAP
1200  XB=X:YB=Y-1
1210  SOUND 1,-15,0,20
1220  FOR K=1 TO 4
1230  PROCBOOM
1240  NEXT K
1250  ENDPROC
1260  DEFPROCFIN
1270  PRINT
1280  IF E=1 THEN PRINT TAB(5,10);"Your
      ship is finished!"
1290  PRINT "You have destroyed ";D;" of
      the enemy":PRINT
1300  IF D<15 THEN PRINT:PRINT "This is
      not enough to save the planet!"
1310  IF D>=15 THEN PRINT "You have
      saved your planet"
1320  PRINT
1330  INPUT "Another Game Y/N",A$
1340  A$=LEFT$(A$,1)
1350  IF A$="N" THEN END
1360  IF A$="Y" THEN RUN
1370  GOTO 1330
1380  ENDPROC
1390  DEF FNS(XS,YS,NS)
1400  YT=YS+SC
1410  YT=YT-INT(ABS(YT)/32)*32
1420  =HIMEM+(YT*40+XS)*8+NS
```

Converted from a program by Simon Bunce

CONQUERING EVEREST

Plan for a Himalayan expedition with the help of your BBC Micro.



This is a computer simulation of an expedition trying to climb Mount Everest. The program is often alarmingly accurate with climbers in high camps, pinned down by the weather and unable to receive supplies from below.

Basically, the idea is a simple one: ropes are laid up the mountain, successively higher camps are occupied and stocked and finally, an attempt on the summit is made. The skill lies in the way supplies are moved around and how the Sherpas and porters are used to the greatest effect.

The expedition consists of 18 climbers, 35 Sherpas (who will carry loads anywhere on the mountain) and 25 porters (who cannot move above Camp 2). There are a total of seven camps on the climb, each of which must be established before moving to the next one. Camp 0 (Base Camp) is the lowest and, at the start of the climb, contains all the people as well as the supplies.

Because It's There

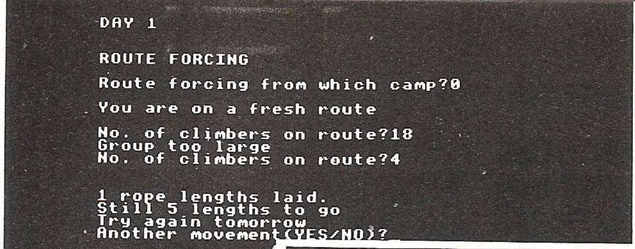
Moving supplies and people about while on the mountain is achieved by defining four types of movement:

1) **CARRY** This is where people carry supplies up to another camp, leave their loads there and return to their original camps. This provides a way of moving loads without moving people and is the type of movement Sherpas and porters are usually used for.

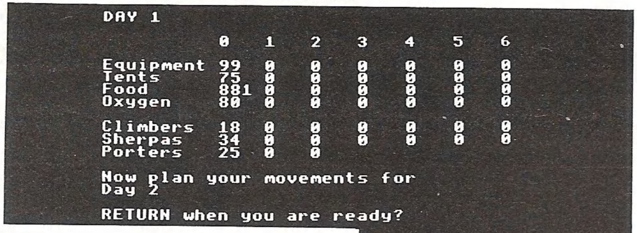
2) **MOVE** This is a similar type of movement to CARRY but here the carrier remains at the camp he carries to. This is used when establishing a new camp or for moving people around lower down on the mountain.

3) **ROUTE** This is the movement used when you are trying to force a route up the mountain.

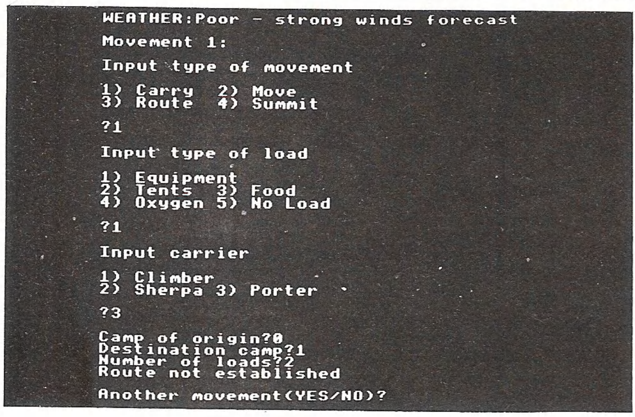
▶



```
DAY 1

ROUTE FORCING

Route forcing from which camp?0

You are on a fresh route

No. of climbers on route?18
Group too large
No. of climbers on route?4


1 rope lengths laid.
Still 5 lengths to go
Try again tomorrow
Another movement(YES/NO)?
```



```
DAY 1

                0   1   2   3   4   5   6

Equipment 99  0   0   0   0   0   0
Tents     75  0   0   0   0   0   0
Food      881 0   0   0   0   0   0
Oxygen    80  0   0   0   0   0   0

Climbers  18  0   0   0   0   0   0
Sherpas   34  0   0   0   0   0   0
Porters   25  0   0

Now plan your movements for
Day 2

RETURN when you are ready?
```



```
WEATHER:Poor - strong winds forecast

Movement 1:

Input type of movement

1) Carry  2) Move
3) Route  4) Summit

?1

Input type of load

1) Equipment
2) Tents  3) Food
4) Oxygen 5) No Load

?1

Input carrier

1) Climber
2) Sherpa 3) Porter

?3

Camp of origin?0
Destination camp?1
Number of loads?2
Route not established

Another movement(YES/NO)?
```

Entering this causes a branch to a separate subroutine where details of the movement are requested. Between each camp are a number of sections all one rope-length long; each section requires one unit of equipment to complete.

The number of rope lengths completed is random but the more people on the route (up to a maximum of four), the more lengths will be laid — the number of sections between each camp can be altered to make the climb easy or difficult. Occasionally, part of the route may be destroyed by an avalanche and have to be repaired so it is often wise to leave some equipment at Camp 0!

4) **SUMMIT** This is a self-explanatory movement used when you have reached Camp 6, although you must remember there are two rope lengths above Camp 6 which must be laid before the summit attempt is made. All the climbers at Camp 6 are assumed to take part but there *must* be at least two. The success of the attempt is dependent on the weather (as are CARRY and MOVE), so you should be prepared to wait at Camp 6 for good weather.

Tactics

At the beginning of the program, there is only one way to go and so there is no need initially to worry about tactics. However, as soon as Camp 1 is reached, a problem arises — how much should you build it up before moving on? The score you get on reaching the summit depends on the number of days it takes you, so fast ascent is desirable. On the other hand, if you move up too quickly the top climbers will soon run out of supplies. Another factor to be taken into account is the distance the Sherpas and porters can move in one day, ie two camps. Obviously, the Sherpas and porters are being used most efficiently when they carry loads up two camps and so it is sensible to build up alternate camps on the climb. By this argument, Camp 1 should not be heavily stocked — although it is a good idea to provide sufficient tents and food for an unexpected descent.

After reaching Camp 2 it is a good idea to pause for a day or two to stock up this base before moving on. All the time climbers are above Camp 2, there should be a daily flow of supplies from Camp 0 to Camp 2. Camp 3 is another relatively unimportant camp, although it does become important later on, since Sherpas can carry loads from Camp 3 directly to Camp 5. Camp 4, and those above it, pose a new problem since oxygen is needed. Oxygen can be used up very quickly and thus it is vital to keep a good supply at Camp 4 for carrying up to higher camps.

At this stage there is a great danger from the 'snowball' effect. This is not a weather feature but occurs when supplies at a high camp are insufficient, forcing the climbers to retreat. They move down to the next camp, and if this camp doesn't have enough supplies, the occupants of this camp are forced to retreat as well, and so on right down the mountain. Often this occurs just as two people are ready to make a summit attempt and they are left stranded while everybody else has been forced down to Camp 2!

If you managed to avoid this trap and are lucky with the weather then you can expect to reach the top in just over a month. It is a very good idea (unless you have a photographic memory) to keep a pencil and paper handy when running the program. You can then note down your planned movements for the next day when all the supplies are printed out at the end of each day.

HOW TO PLAY

Typing 'RUN' produces the heading 'Day 1', gives a weather report for the day ahead and asks for a movement input. Since, at this stage, the only way to go is up, 'ROUTE' should be entered. The program then branches to the 'ROUTE' subroutine where you will be asked to enter the camp you are climbing from and the number of people in the group. The number of rope lengths laid is then printed out and the program branches back to the main body where you are asked if you want to make another movement that day. Entering 'NO' signals the end of that day and a series of status reports are produced.

First, the day's movements

Flexi Study
-NEC-
RECOMMENDED TEXT ®

30 Hour BASIC

Clive Prigmore

BBC
NEC
ENROL TODAY

BASIC for the Computer Programme (BBC TV)

National Extension College Correspondence Text

What is BASIC?

Microcomputers are the tool of the 80's. BASIC is the language that all of them use. So the sooner you learn BASIC, the sooner you will understand the microcomputer revolution.

30 Hour BASIC is a simple self-instructional course on the language of microcomputers. But programs need more than language: they need structure as well. So the course also teaches you good programming techniques. You'll learn how to keep, order and sort files, records and directories; how to print letters and addresses; how to invent your own computer games; how to handle numbers and so on.

Can anyone take the course?

Yes. You don't need a microcomputer to do the course: you can study it by yourself or as an NEC correspondence student. Used without a computer it will take you about 30 hours. But you will get more out of it if you have your own microcomputer or access to one. That way you'll need to allow extra time for keying and running programs.

Which computer?

30 Hour BASIC has been specially prepared for **The Computer Programme** (BBC TV) but can be used with any microcomputer except the ZX80. (If you are going to use a ZX81 you will need the special Z81 Edition of the course which we will supply—make sure you tell us which edition overleaf.)

For use with the standard edition of **30 Hour BASIC,** we will supply free an extra sheet of notes for most other microcomputers (for 380Z and 480Z users — an extra Guide can be obtained free of charge from Research Machines Ltd., Mill Street, Oxford). Make sure you tell us you need the extra free sheet — *see* Order Form.

What does a program look like?

Here's a simple example: a personal telephone directory to type into your computer.

```
 10 REM**TELEPHONE DIRECTORY**
 15 CLEAR 100 (not needed on all computers)
 20 INPUT "NAME OF PERSON SOUGHT"; Q$
 30 READ N$,T$
 40 IF N$="ZZZZ" THEN 70
 50 IF Q$=N$ THEN 90
 60 GOTO 30
 70 PRINT Q$ "IS NOT IN FILE"
 80 STOP
 90 PRINT Q$; " 'S NUMBER IS "; T$
100 STOP
```

Then you add your own personal data.

```
200 DATA BILL, 45335
210 DATA MARY, 74558
           etc for up to 100 names.
300 DATA ZZZZ, 1234
```

And now to use it.

```
RUN
NAME OF PERSON SOUGHT? MARY
MARY'S NUMBER IS 74558
```

As you can see, programs are mostly in simple English with a few abbreviations and symbols. This helps you to learn to write programs quickly and easily. **30 Hour BASIC** shows you step by step how to build up programs like this. At the end of the course you will have an extensive library of useful and interesting programs.

Cassettes

If you are a slow typist or just short of time, you may not want to key in all the programs in the course. To save you the bother, you can buy from NEC two cassettes of all the main programs in the course. Price £5.98 each. These can be ordered in section B of page 4. (NB These cassettes are suitable for the BBC Microcomputer only.)

Certificate in Computer Literacy

The National Extension College has made arrangements with the City & Guilds of London Institute (CGLI) for the CGLI to offer a new assessment to lead to a Certificate in Computer Literacy. It will be possible to sit for this assessment at centres throughout the UK. For details write to Mr A. N. A. Ward, City & Guilds of London Institute, 46 Britannia Street, LONDON WC1X 9RG.

How to enrol

You can study the course in one of three ways:

1. By enrolling as an NEC correspondence student. You do this by completing section A on page 4. We will then send you the course and appoint one of our correspondence tutors to you to mark and comment on your assignments. There are nine assignments in the course.
2. You may be able to take the course at a local FlexiStudy centre. Try your local college of further education and your local education authority. NEC is preparing a list of centres offering **30 Hour BASIC.** Write for details if necessary.
3. You may wish to study the course by yourself without the help of a tutor. To do this, simply order the course text using section B of page 4.

Course contents

Introduction

UNIT 1 Simple statements and commands

1.1 What does a computer do?, 1.2 What is a computer?, 1.3 What is BASIC?, 1.4 A simple problem, 1.5 Statement numbers, 1.6 Execution and commands, 1.7 Execution and data, 1.8 INPUT, PRINT and LET, 1.9 Store locations, 1.10 Copying and overwriting, 1.11 Arithmetic operators, 1.12 Numerical constants, 1.13 The remark statement: REM, 1.14 More complicated arithmetic, 1.15 Literal printing.

UNIT 2 Making decisions

2.1 Introduction, 2.2 PRINT . . ., 2.3 Repetitions and GOTO, 2.4 Programming style, 2.5 IF . . . THEN . . ., 2.6 Inequalitites, 2.7 Flowcharts, 2.8 Counting, 2.9 Comparisons.

UNIT 3 Strings

3.1 What is a string?, 3.2 More about strings, 3.3 PRINT . . .; . . ., 3.4 INPUT ". . ."; . . ., 3.5 Numbers and strings in PRINT statements, 3.6 Standard letters, 3.7 Patterns, files, READ with DATA, 3.8 Sorting. Appendix.

UNIT 4 Lists

4.1 Variables, 4.2 Lists, 4.3 List variables, 4.4 List input and output, 4.5 The FOR . . . NEXT . . . loop, 4.6 Nested loops, 4.7 Interchanging.

UNIT 5 An end to strings and PRINT

5.1 Introduction, 5.2 Length of a string of characters, 5.3 Frequency tables, 5.4 Frequency diagrams, 5.5 Tabulation, 5.6 Cutting up strings, 5.7 VAL.

UNIT 6 Mainly about dice and games

6.1 Random numbers, 6.2 The RND function, 6.3 RND on the BBC Computer, 6.4 Random number postcript, 6.5 Two examples, 6.6 Keeping scores, 6.7 Short cuts in program writing, 6.8 Concatenation, 6.9 STR$.

UNIT 7 Handling numbers

7.1 Introduction, 7.2 Averages and the arithmetic mean, 7.3 Range, 7.4 Number crunching, 7.5 Dry running, 7.6 The representation of numbers, 7.7 The INT-function for rounding, 7.8 The ABS-function, 7.9 Iteration.

UNIT 8 Introduction to data processing

8.1 Introduction, 8.2 Sorting, 8.3 Subroutines, 8.4 Searching, 8.5 Tables.

UNIT 9 File-handling (NOT INCLUDED IN THE ZX81 EDITION)

9.1 Saving programs, 9.2 LPRINT, 9.3 Sequential files, 9.4 Create and access, 9.5 Files in flowcharts, 9.6 Sorting filed data, 9.7 Merge, 9.8 Delete.

INDEX

Each unit contains self-assessment questions with full answers. At the end of each unit there is an assignment for tutor marking.

A

CORRESPONDENCE COURSE ENROLMENT FORM

Fill in this form and post it to the National Extension College, 18 Brooklands Avenue, Cambridge CB2 2HN. We will post your course and details of your tutor to you. You can start the course as soon as you get it and, when you are ready, post your first assignment to your tutor.

What does it cost?
The full fee for **30 Hour BASIC** with tuition is £36 which covers your course material and tuition. (The cassette tapes are not included in this fee. If you want the tapes as well, please order them separately.) If you wish to pay the course fee by instalments you may do so with a deposit of £14 and five monthly instalments of £5 (total £39).

Conditions of enrolment
As soon as your enrolment is accepted, the course will be sent to you. If within six weeks of the date of despatch (two months if you live abroad), you decide that the course is too difficult or unsuitable for any reason, you may write to us and cancel your enrolment. You would then receive back the full amount of the fees you have paid less an administrative charge of 25% of the full cash fee, plus a charge for any assignments which you have had marked. After this period has elapsed you would NOT be eligible for a cash refund and, on cancelling your course, NEC would give you a transfer right for the fees paid less an administrative and assignment charge (see above). This transfer right can be used towards the fee when enrolling on any other NEC course.

FOR USE UP TO 1 SEPTEMBER, 1982. Price and Fee revisions expected

PLEASE COMPLETE IN BLOCK LETTERS

Title and initials

Forenames

Surname

Address

Postcode

Tel. No.

Office use only

Student No.

Tutor No.

NB
Overseas students: please add extra for overseas postage – £7 Europe; £8 outside Europe.

Date of birth day month year

Occupation

Have you enrolled on an NEC course before? yes ☐ no ☐

Previous student number (if known)

Will you own/have access to a microcomputer when you are taking this course?

If you will, please state the name and model

PAYMENT

Please complete one of (a), (b), (c) or (d) below.

(a) Full payment now ☐ (tick)
I wish to pay the full fee now and enclose a cheque/PO for £36.

(b) Credit card ☐ (tick)
I authorise you to debit may Access/Barclaycard/Trustcard account with my tuition fee of £36

Access Account No.

Barclaycard Account No.

Trustcard Account No.

(c) Cash instalments ☐ (tick)
I wish to pay by cash instalments and enclose a cheque/PO for the deposit of £14 and agree to pay five monthly instalments of £5.

(d) Bankers' Order ☐ (tick)

BANKERS STANDING ORDER FORM

(Fill in the name and address of your bank in block capitals.) TO: THE MANAGER

Please pay Midland Bank Ltd., 465 Bethnal Green Road, London E2 (40–01–18) for the credit of the National Extension College Trust Ltd. A/C No. 70614939.

the sum of £14 commencing on the fifth day of 1982, and on the same day of each following calendar month for five months the sum of £5 AND CHARGE SUCH PAYMENTS TO MY ACCOUNT.

Yours faithfully, SIGNATURE Current A/C No.

Name
(block letters please)

Address

NEC Ref.No.

Signature Date

- -

B ORDER FORM: CASSETTES and/or SELF-INSTRUCTIONAL COURSE

Please supply: £

. . . . .copies of **30 Hour BASIC** without tuition, standard edition, @ £5.50.

. . . . .copies of **30 Hour BASIC** without tuition, ZX81 edition,@ £5.50

. . . . .copies of cassette 1 @ £5.98 inc VAT (BBC Computer only)

. . . . .copies of cassette 2 @ £5.98 inc VAT (BBC Computer only)

for free sheet, my microcomputer is a . . . .

. . . . Linex Flow Chart Template (if wanted, not essential) £3.50 inc. VAT

Cheque/PO enclosed for:
(prices inc p&p in the UK) TOTAL

Name (BLOCK LETTERS) . . . .

Address . . . .

. . . .

- -

Send to: National Extension College, 18 Brooklands Avenue, Cambridge CB2 2HN.
GROUP TRAINING SCHEMES AVAILABLE – WRITE FOR DETAILS.

Inspection copies for College use available.

5/82

are printed out, those not achieved (the success of a CARRY depends on the weather, although those made by climbers are always successful) being indicated by '!NC!'. The next status report is a supply check which gives a clear indication if any camp is not sufficiently stocked with tents or consumables (in extreme cases the occupants will retreat to the camp below them). Finally, a comprehensive picture of the situation on the mountain is given, displaying the contents and occupants of each camp.

The program then starts a new day and the process begins again. The number of possible movements increases very rapidly as you progress up the mountain so you need to develop a strategy for your movements later in the program.

The rules of the game are as follows:

1) At the start of each game all people and supplies are at Camp 0.
2) A maximum of 10 movements are allowed per day.
3) At the end of each day, all the day's movements are printed out and those which were not actually 'carried' are marked by '!NC!'.
4) Tents usually hold two people but can hold up to three.
5) One unit of food lasts four man-days.
6) One unit of oxygen lasts one man-day at Camp 6 and two man-days at Camps 4 and 5.
7) The success of a CARRY depends on the height and the weather, although when the weather is 'perfect conditions', all 'carries' are completed.
8) Before any new camp site is established, fixed ropes must be laid from the camp below.
9) Only climbers can force the route up the mountain.
10) A maximum of four climbers are allowed to force the route at any one time.
11) Only one attempt per day is allowed on the route above any one camp.
12) All occupied camps must have sufficient supplies for the people in them at all times.
13) Nobody can move more than two camps up or down in any one day.
14) Climbers need oxygen at Camps 4 and above.
15) Sherpas need oxygen at Camps 5 and above.
16) Porters are not able to move above Camp 2.

SCOPE FOR ALTERATION

To make the climb easier or more difficult, you can alter the number of sections between each camp.

PROGRAM STRUCTURE

Statement	Function	Action
Line 10	Set Up	Calls routine to set up arrays and initialise variables.
Lines 20-180	Daily Movements	Counts days since the start of the climb. Calls weather routine and asks user to choose which type of movement is to be done.
Lines 190-470	Load	Asks user to choose type of load, type of carrier, origin and destination of loads and number of them.
Lines 480-510	Extra Movements	Gives user the option of further movements.
Lines 520-580	Messages	Calls routines to display information from top line of screen.
Lines 590-630	Height	Works out highest point reached.
Lines 640-690	Reset Arrays	Resets arrays for the six camps.
Lines 700-850	Weather	Selects weather conditions at random and displays them.
Lines 860-1070	Status	Displays a table of information regarding supplies at camps.
Lines 1080-1290	Move Logic	Calculates the outcomes of the movements selected.
Lines 1300-1640	Carry	If CARRY is specified, checks for legality of move.
Lines 1650-2080	Route Forcing	Input, logic and messages for route-forcing option.
Lines 2090-2220	Summit Bid	Logic and messages for attempt to reach the summit.
Lines 2230-2300	End Game	Allows option of another game and prints score.
Lines 2310-2560	Avalanche	Generates avalanches at random and displays messages.
Lines 2570-2690	Consumables	Calculates consumption of commodities.
Lines 2700-2900	Supply Checks	Checks whether sufficient supplies remain and prints messages.
Lines 2910-2970	Retreat	Prints message and does calculations in event of a retreat.
Lines 2980-3250	Summary	Displays a summary statement of the day's movements — any not achieved are denoted '! NC !'.
Lines 3260-3360	Data Arrays	Sets up initial data arrays.

▶

```
  10  GOSUB 3270
  20  REM**INPUT
  30  CLS:DD=DD+1
  40  TL$="DAY "+STR$(DD):GOSUB 560
  50  NM=0:RF=-1
  60  GOSUB 650:GOSUB 710:GOSUB 2320
  70  IF W=-1 THEN 520
  80  NM=NM+1
  90  IF NM<=10 THEN 110
 100  PRINT "No more carries":GOTO 520
 110  PRINT:PRINT"Movement ";NM;":":PRINT
 120  PRINT"Input type of movement"
 130  PRINT
 140  PRINT"1) Carry","2) Move"
 150  PRINT"3) Route","4) Summit"
 160  PRINT
 170  INPUT IN$
 180  M1=VAL(IN$)
 190  IF M1<1 OR M1>4 THEN 170
 200  IF M1=3 THEN GOSUB 1660:GOTO 480
 210  IF M1=4 THEN GOSUB 2100:GOTO 480
 220  PRINT:PRINT"Input type of load":PRINT
 230  PRINT"1) Equipment"
 240  PRINT"2) Tents","3) Food"
 250  PRINT"4) Oxygen","5) No Load"
 260  PRINT
 270  INPUT IN$:M2=VAL(IN$)
 280  IF M2<1 OR M2>5 THEN 270
 290  PRINT
 300  PRINT"Input carrier":PRINT
 310  PRINT "1) Climber"
 320  PRINT"2) Sherpa","3) Porter"
 330  PRINT
 340  INPUT IN$:M3=VAL(IN$)
 350  IF M3<1 OR M3>3 THEN 340
 360  PRINT
 370  INPUT"Camp of origin",IN$:M4=VAL(IN$)
 380  IF M4<0 OR M4>6 THEN 370
 390  IF M4=0 AND IN$<>"0" THEN 370
 400  INPUT"Destination camp",IN$:M5=VAL(IN$)
 410  IF M5<0 OR M5>6 THEN 400
 420  IF M5=0 AND IN$<>"0" THEN 400
 430  INPUT"Number of loads",IN$:M6=VAL(IN$)
 440  IF M6<0 THEN 430
 450  IF M6=0 AND IN$<>"0" THEN 430
 460  GOSUB 1090
 470  PRINT
 480  INPUT"Another movement(YES/NO)",RM$
 490  RM$=LEFT$(RM$,1)
 500  IF RM$="Y" THEN CLS:GOSUB 560:GOTO 80
 510  IF RM$<>"N" THEN 480
 520  GOSUB 2580:GOSUB 2990:GOSUB 2710
 530  GOSUB 870:GOTO 30
 540  REM**UTILITY
 550  REM**Top Line
 560  PRINT TAB(0,1);TL$
 570  PRINT
 580  RETURN
 590  REM**High Point
 600  D=7
 610  D=D-1:IF TN(D)<>0 THEN HP=D:RETURN
 620  IF D<=0 THEN RETURN
 630  GOTO 610
 640  REM** Array resets
 650  FOR D=0 TO 6
 660  DC(D)=CL(D):DS(D)=SH(D):NEXT
 670  FOR D=0 TO 2:DP(D)=PT(D):NEXT
 680  FOR D=0 TO 10:M$(D)="":NEXT
 690  RETURN
 700  REM**WEATHER
 710  PRINT"WEATHER:";
 720  W=INT(RND(1)*3+1.4)
 730  ON W GOTO 740,800,820,840
 740  IF RND(1)>.85 THEN 770
 750  PRINT"Very bad - snow & wind forecast"
 760  RETURN
 770  PRINT"! STORM ! - No movement today"
 780  INPUT"Press RETURN to continue",Z$
 790  W=-1:RETURN
 800  PRINT"Poor - strong winds forecast"
 810  RETURN
 820  PRINT"Good - only light snow predicted"
 830  RETURN
 840  PRINT"Perfect conditions"
 850  RETURN
 860  REM**DISPLAY
 870  CLS:GOSUB 560
```

```
 880  FOR D=0 TO 6:PRINT TAB(10+4*D);D;:NEXT
 890  PRINT:PRINT:PRINT"Equipment";
 900  FOR D=0 TO 6:PRINT TAB(10+4*D);EQ(D);:
      NEXT
 910  PRINT:PRINT"Tents";
 920  FOR D=0 TO 6:PRINT TAB(10+4*D);
      TN(D);:NEXT
 930  PRINT:PRINT"Food";
 940  FOR D=0 TO 6:PRINT TAB(10+4*D);FD(D);:
      NEXT
 950  PRINT:PRINT"Oxygen";
 960  FOR D=0 TO 6:PRINT TAB(10+4*D);OX(D);:
      NEXT
 970  PRINT:PRINT:PRINT"Climbers";
 980  FOR D=0 TO 6:PRINT TAB(10+4*D);CL(D);:
      NEXT
 990  PRINT:PRINT"Sherpas";
1000  FOR D=0 TO 6:PRINT TAB(10+4*D);SH(D);:
      NEXT
1010  PRINT:PRINT"Porters";
1020  FOR D=0 TO 2:PRINT TAB(10+4*D);PT(D);:
      NEXT
1030  PRINT:PRINT
1040  PRINT"Now plan your movements for"
1050  PRINT"Day ";DD+1:PRINT
1060  INPUT"RETURN when you are ready",Z$
1070  RETURN
1080  REM**MOVE, CARRY
1590  M$(NM)=" "+STR$(M1)+" "+STR$(M2)
1100  M$(NM)=M$(NM)+" "+STR$(M3)+" "+STR$(M4)
1110  M$(NM)=M$(NM)+" "+STR$(M5)+" "+STR$(M6)
1120  ME=0:GOSUB 1310:IF ME=2 THEN RETURN
1130  IF ME=1 THEN NM=NM-1:RETURN
1140  ON M2 GOTO 1150,1170,1190,1210,1220
1150  EQ(M4)=EQ(M4)-M6:EQ(M5)=EQ(M5)+M6
1160  GOTO 1220
1170  TN(M4)=TN(M4)-M6:TN(M5)=TN(M5)+M6
1180  GOTO 1220
1190  FD(M4)=FD(M4)-M6:FD(M5)=FD(M5)+M6
1200  GOTO 1220
1210  OX(M4)=OX(M4)-M6:OX(M5)=OX(M5)+M6
1220  IF M1=1 THEN RETURN
1230  ON M3 GOTO 1240,1260,1280
1240  CL(M4)=CL(M4)-M6:CL(M5)=CL(M5)+M6
1250  RETURN
1260  SH(M4)=SH(M4)-M6:SH(M5)=SH(M5)+M6
1270  RETURN
1280  PT(M4)=PT(M4)-M6:PT(M5)=PT(M5)+M6
1290  RETURN
1300  REM**CHECK 1
1310  IF M3<>3 OR M5<=2 THEN 1330
1320  PRINT"Porters too high":ME=1
1330  ON M2 GOTO 1340,1360,1380,1400,1420
1340  IF EQ(M4)-M6>=0 THEN 1420
1350  GOTO 1410
1360  IF TN(M4)-M6>=0 THEN 1420
1370  GOTO 1410
1380  IF FD(M4)-M6>=0 THEN 1420
1390  GOTO 1410
1400  IF OX(M4)-M6>=0 THEN 1420
1410  PRINT"Too few supplies for carry":ME=1
1420  ON M3 GOTO 1430,1450,1470
1430  IF DC(M4)-M6<0 THEN 1490
1440  DC(M4)=DC(M4)-M6:GOTO 1500
1450  IF DS(M4)-M6<0 THEN 1490
1460  DS(M4)=DS(M4)-M6:GOTO 1500
1470  IF DP(M4)-M6<0 THEN 1490
1480  DP(M4)=DP(M4)-M6:GOTO 1500
1490  PRINT"Too few carriers for carry":ME=1
1500  IF M5-M4<=2 THEN  1520
1510  PRINT"Too much height gain":ME=1
1520  IF M5-M4<>2 THEN 1540
1530  IF RF(M4+1,1)<>RF(M4,0) THEN 1550
1540  IF RF(M4,0)=RF(M4,1) OR M4>M5 THEN 1560
1550  PRINT"Route not established":ME=1
1560  IF ME=1 THEN RETURN
1570  IF W=4 THEN RETURN
1580  IF M3=1 THEN RETURN
1590  WF=RND(1)*(W+RND(1))
1600  IF M5<=2 AND WF>.8 THEN RETURN
1610  IF M5<=4 AND WF>1 THEN RETURN
1620  IF M5<=6 AND WF>1.25 THEN RETURN
1630  M$(NM)=M$(NM)+"NC"
1640  ME=1:RETURN
1650  REM**ROUTE
1660  CLS:GOSUB 560
1670  M$(NM)=" 3"
```

```
1680  PRINT:PRINT"ROUTE FORCING":PRINT
1690  PRINT"Route forcing from which camp";
1700  INPUT IN$:M4=VAL(IN$)
1710  IF M4>6 OR M4<0 THEN 1700
1720  IF M4=0 AND IN$<>"0" THEN 1700
1730  IF RF<>M4 THEN 1750
1740  PRINT"One try per day":RETURN
1750  IF EQ(M4)>0 THEN 1770
1760  PRINT"No equipment available":RETURN
1770  IF RF(M4,0)<>RF(M4,1) THEN 1790
1780  PRINT"Route already completed":RETURN
1790  PRINT:RF=M4
1800  IF RF(M4,1)<>0 THEN 1830
1810  PRINT"You are on a fresh route"
1820  GOTO 1850
1830  PRINT TAB(0);RF(M4,0)-RF(M4,1);"
      rope lengths to"
1840  PRINT"go to finish route"
1850  PRINT
1860  INPUT"No. of climbers on route",IN$
1870  M6=VAL(IN$):IF M6<=0 THEN 1860
1880  IF CL(M4)<>0 THEN 1900
1890  PRINT"No climbers available":
      RETURN
1900  IF CL(M4)-M6>=0 THEN 1930
1910  PRINT"Only ";CL(M4);" climbers available"
1920  GOTO 1860
1930  IF M6<=4 THEN 1950
1940  PRINT"Group too large":GOTO 1860
1950  PRINT
1960  RT=INT(RND(1)*M6+1)
1970  IF RT>EQ(M4) THEN RT=EQ(M4)
1980  EQ(M4)=EQ(M4)-RT
1990  RF(M4,1)=RF(M4,1)+RT
2000  IF RF(M4,1)>=RF(M4,0) THEN 2050
2010  PRINT
2020  PRINT TAB(0);RT;" rope lengths laid."
2030  PRINT"Still ";RF(M4,0)-RF(M4,1);
      "lengths to go"
2040  PRINT"Try again tomorrow":RETURN
2050  RF(M4,1)=RF(M4,0)
2060  PRINT"Congratulations"
2070  PRINT"You have completed this section"
2080  RETURN
2090  REM**SUMMIT
2100  CLS:GOSUB 560
2110  M$(NM)=" 4"
2120  PRINT" SUMMIT BID":PRINT
2130  IF RF(6,1)=2 THEN 2150
2140  PRINT"Top rope sections not laid":RETURN
2150  IF CL(6)>=2 THEN 2170
2160  PRINT "Too few climbers for bid":RETURN
2170  IF RND(1)*W>2.5 THEN 2200
2180  PRINT "Bid unsuccessful -";
2190  PRINT"Try again tomorow":RETURN
2200  PRINT:SR=SR+CL(6)
2210  PRINT" !!! SUMMIT REACHED !!"
2220  PRINT:PRINT"Congratulations":PRINT
2230  PRINT "Do you want to make another ";
2240  PRINT"attempt";
2250  INPUT" (YES/NO)",D$
2260  IF D$<>"N" THEN GOTO 2250
2270  IF D$="Y" THEN RETURN
2280  S=(SR*1000)/DD:PRINT
2290  PRINT"Your score is";S
2300  END
2310  REM** ACCIDENT
2320  IF RND(1)<.9 THEN RETURN
2330  RAN=INT(RND(1)*1.4+1)
2340  ON RAN GOTO 2360,2490
2350  REM** ICE FALL
2360  GOSUB 600:IF HP<1 THEN RETURN
2370  RAN=INT(RND(1)*2+1.5)
2380  RF(0,1)=RF(0,1)-RAN
2390  PRINT"An avalanche has swept away ";
2400  PRINT"part of the route between"
2410  PRINT"camps 0 & 1."
2420  PRINT TAB(0);RAN;" lengths ";
2430  PRINT"of rope have been lost."
2440  PRINT"This must be repaired before"
2450  PRINT"any loads can be moved from";
2460  PRINT" Camp 0."
2470  RETURN
2480  REM**AVALNCHE
2490  GOSUB 600:AC=INT(RND(1)*HP)
2500  PRINT "Camp ";AC;" has been hit by an"
2510  PRINT "avalanche"
2520  RAN=INT(RND(1)*2+1.6)
2530  PRINT TAB(0);RAN;
2540  PRINT" tents have been lost."
2550  TN(AC)=TN(AC)-RAN
2560  RETURN
2570  REM**CONSUMABLES
2580  FOR D=0 TO 6
2590  TP=CL(D)+SH(D)
2600  IF D<=2 THEN TP=TP+PT(D)
2610  IF TP=0 THEN NEXT:RETURN
2620  IF D=4 THEN OX(D)=OX(D)-CL(D)/2
2630  IF D=5 THEN OX(D)=OX(D)-TP/2
2640  IF D=6 THEN OX(D)=OX(D)-TP
2650  FD(D)=FD(D)-INT(TP/4+.5)
2660  IF OX(D)<0 THEN OX(D)=0
2670  IF FD(D)<0 THENFD(D)=0
2680  NEXT
2690  RETURN
2700  REM *CHECK 2
2710  CLS:GOSUB 560:PRINT
2720  PRINT"Supply Check":PRINT
2730  FOR D=0 TO 6 :EV=0
2740  TP=CL(D)+SH(D)
2750  IF D<=2 THEN TP=TP+PT(D)
2760  IF TP=0 THEN NEXT:INPUT D$:RETURN
2770  IF TP<=2*TN(D) THEN 2800
2780  PRINT"Too few tents at camp ";D
2790  IF TP>4*TN(D) THEN EV=1
2800  IF TP<4*FD(D) THEN 2830
2810  PRINT"Food is running out at camp";D
2820  IF TP>6*FD(D) THEN EV=1
2830  IF D<4 THEN 2870
2840  IF TP<2*OX(D) THEN 2870
2850  PRINT"Oxygen is running out at camp";D
2860  IF TP>3*OX(D) THEN EV=1
2870  IF EV=1 THEN GOSUB 2920:GOTO 2890
2880  PRINT "Everything OK at camp ";D
2890  NEXT
2900  INPUT D$:RETURN
2910  REM*RETREAT
2920  PRINT:PRINT"!!! The climbers at camp";D
2930  PRINT"are retreating!!":PRINT
2940  CL(D-1)=CL(D-1)+CL(D):CL(D)=0
2950  SH(D-1)=SH(D-1)+SH(D):SH(D)=0
2960  IF D<=2 THEN PT(D-1)=PT(D-1)+PT(D):
      PT(D)=0
2970  RETURN
2980  REM*PRINT CARRY
2990  CLS:GOSUB 560
3000  PRINT:PRINT"The days movements:-":PRINT
3010  FOR D=1 TO NM
3020  PRINT TAB(0);D;":";
3030  IF FNM(2)>4 OR FNM(2)<1 THEN GOTO 3230
3040  ON FNM(2) GOTO 3050,3060,3070,3080
3050  PRINT"carry,";:GOTO 3090
3060  PRINT "Move,";:GOTO 3090
3070  PRINT"Route";:GOTO 3230
3080  PRINT"Summit":GOTO 3230
3090  ON FNM(4) GOTO 3100,3110,3120,3130,3140
3100  PRINT"Equipment,";:GOTO 3150
3110  PRINT"Tents,";:GOTO 3150
3120  PRINT"Food,";:GOTO 3150
3130  PRINT"Oxygen,";:GOTO 3150
3140  PRINT"No load,";
3150  ON FNM(6)GOTO 3160,3170,3180
3160  PRINT"Climber,";:GOTO 3190
3170  PRINT"Sherpa,";:GOTO 3190
3180  PRINT"Porter,";
3190  PRINT FNM(8);"->";FNM(10);",";FNM(12);
3200  IF RIGHT$(M$(D),2)="NC" THEN 3220
3210  PRINT:GOTO 3230
3220  PRINT "! NC !"
3230  NEXT
3240  PRINT:PRINT:PRINT"Presss RETURN
      to continue";
3250  INPUT Z$:RETURN
3260  REM**INIIAL
3270  DIM EQ(6),TN(6),FD(6),OX(6)
3280  DIM CL(6),SH(6),PT(6)
3290  DIM DC(6),DS(6),DP(6)
3300  DIM RF(6,1),M$(10)
3310  EQ(0)=100:TN(0)=75:FD(0)=900:OX(0)=80
3320  CL(0)=18:SH(0)=34:PT(0)=25
3330  FOR D=0 TO 6:READ RF(D,0):NEXT
3340  DEF FNM(KD)=VAL(MID$(M$(D),KD,2))
3350  RETURN
3360  DATA 6,5,4,3,4,3,2
```

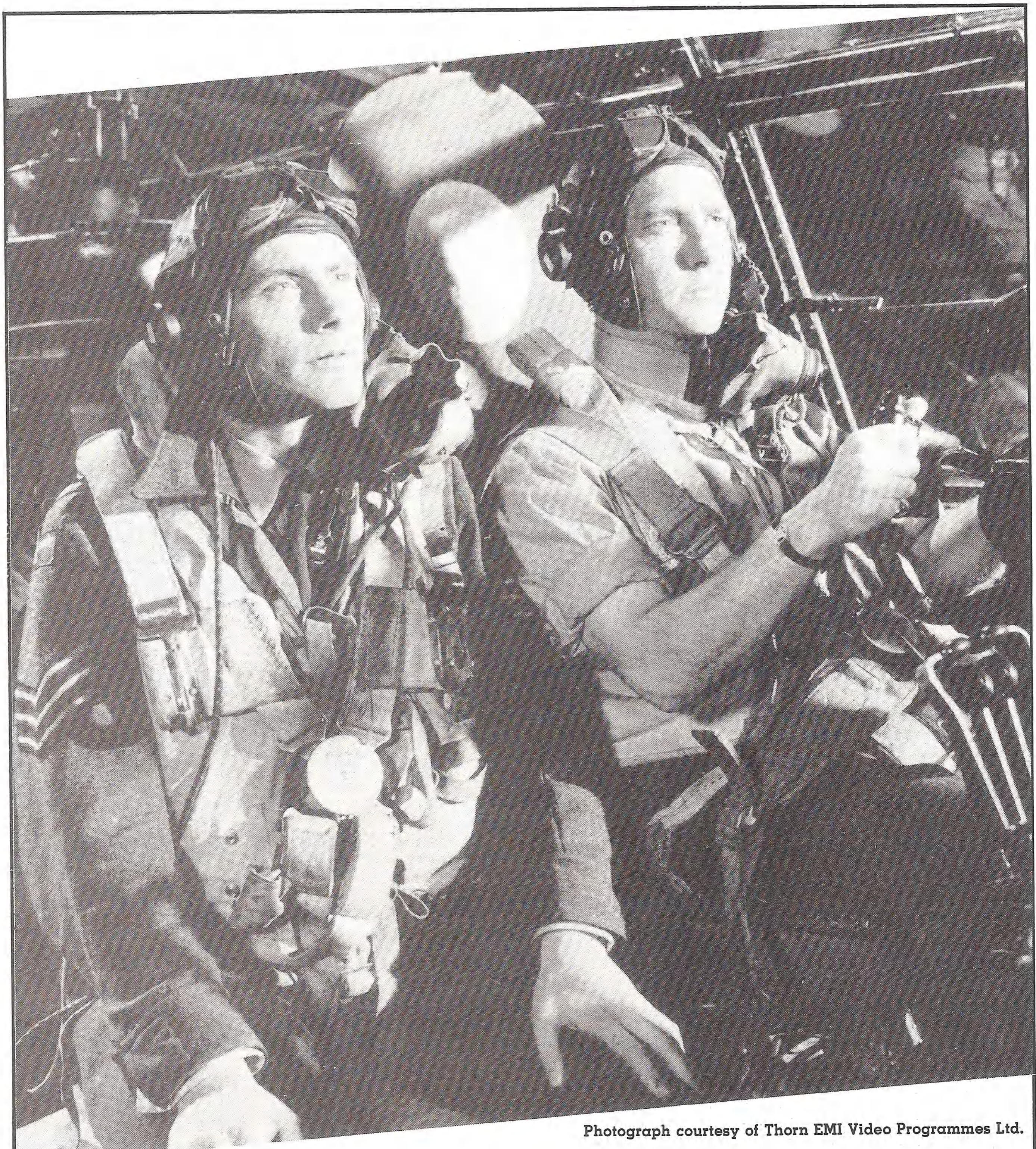
Converted from a program by S Draper

DAMBUSTER

Use your judgement to destroy a dam with a bouncing bomb.

Barnes Wallis's bouncing bomb and the daring air crews who risked flying deep into enemy territory to bombard the Ruhr and so interrupt the Nazis' armament industry are more than history. They are a twentieth century legend.

So many aspects of the enterprise catch the imagination. The ingenious design of the bomb was the initial marvel. But the skill and courage required to make them



Photograph courtesy of Thorn EMI Video Programmes Ltd.

effective was even more remarkable. Only if the bomb struck the water at the correct height and distance from the dam would it actually destroy its target.

The airmen faced multiple dangers: the flight levels required for dropping the bombs were so low they ran the risk of crashing; they flew for many miles inside enemy lines thus risking being shot down; and they were flying to the limits of their fuel capacity and so had to take care to conserve fuel in order to return safely to base.

This program simulates a dambusting raid — but without the additional risks of facing enemy fire or running out of fuel.

TECHNICAL DETAILS

This is a straightforward conversion from PET BASIC. The procedure PROCXY is used to convert PET screen address POKEs to BBC BASIC PRINT TAB(X,Y);CHR$(C); using the usual method. The only PET graphics characters required are produced in lines 10-40.

HOW TO PLAY

The screen displays a reservoir with a dam across it. The objective is to destroy the dam. You have three aircraft, each of which carries a single bomb.

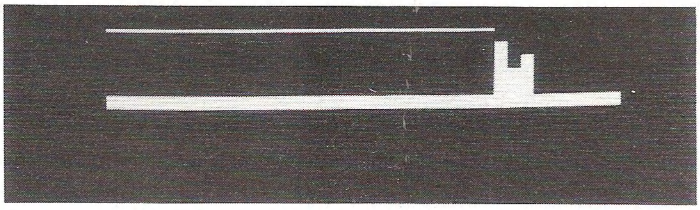
The initial height of the aircraft is set at random but you can adjust it by pressing 'U' (to make it go UP) or 'D' (to make it go DOWN). Take care though — if you fly too low you may crash. To drop your bomb press 'B'. It will then fall in a parabolic arc towards the water and will bounce along the surface of the water until the maximum height of the bounce drops too low.

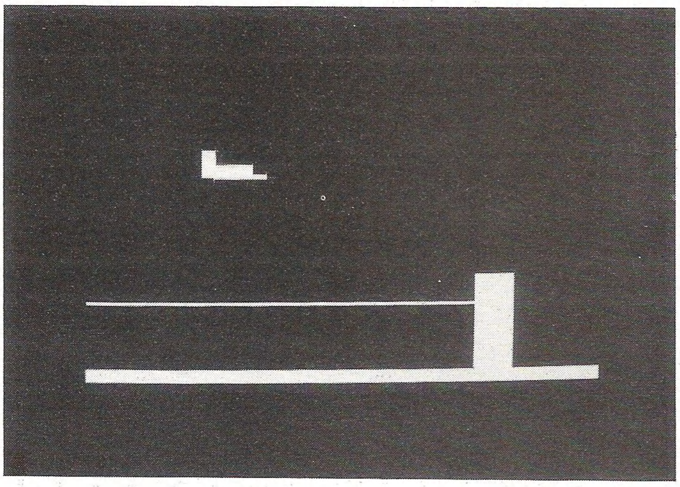
If you hit the dam with your bomb you will destroy it. After your first aircraft has crashed or completed its mission, the second will appear. When all three aircraft have been used, your score (the number of dams destroyed and the number of aircraft crashed) will be displayed.

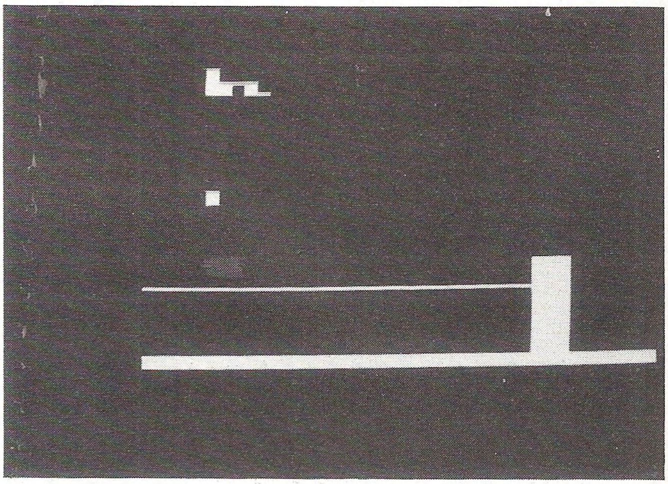
PROGRAM STRUCTURE

Statement	Function	Action
Lines 10-50	Set Up	Selects MODE 4. Defines graphics characters.
Lines 60-110	Initialise	Defines arrays used and sets initial values of variables.
Lines 120-220	Plot Dam	Plots screen display of 'reservoir' and 'dam'.
Lines 230-420	Aircraft	Sets initial position of aircraft at random height. Checks height, issues warning and registers crash if drops too low.
Lines 430-510	Action	Gets keypress and responds by alterting height of aircraft or dropping 'bomb'.
Lines 520-650	Bomb Logic	Causes bomb to drop in parabolic arc and bounce.
Lines 660-860	Dambust	Destroys dam.
Lines 870-1000	Low bomb	Prints bomb splash.
Lines 1010-1060	End Game	Prints score message and asks if another game is wanted. If so re-runs, if not ends.
Lines 1070-1110	Address	Converts PET screen address to x and y co-ordinates and prints character.

Photograph courtesy of Thorn EMI Video Programmes Ltd.





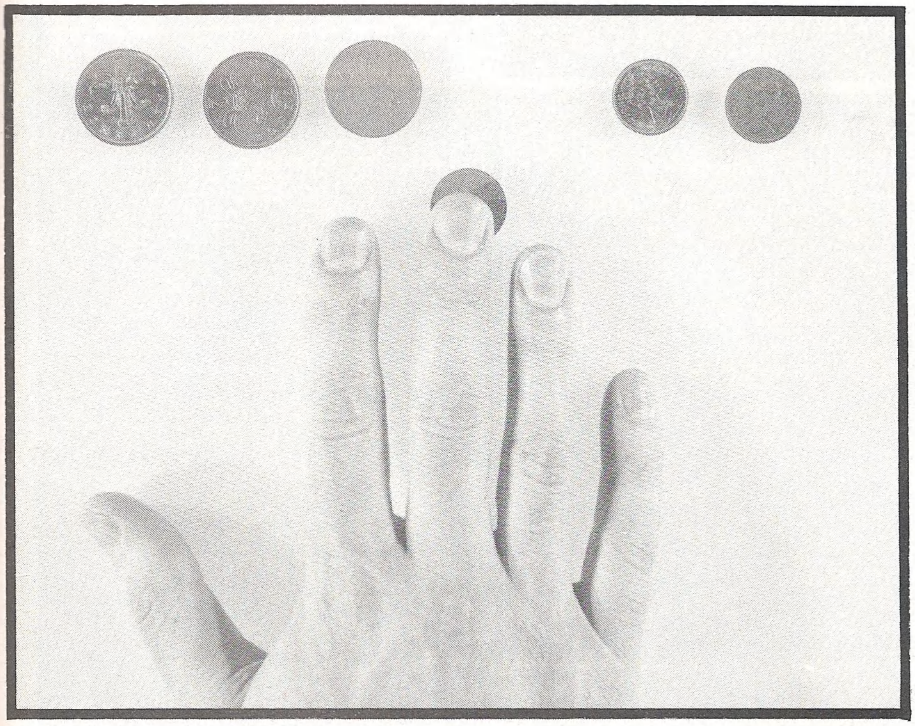


```
  10  MODE 4
  20  VDU 23,224,&FF,&FF,&FF,&FF,&FF,&FF,
      &FF,&FF
  30  VDU 23,225,&FF,&FF,0,0,0,0,0,0
  40  VDU 23,226,0,0,0,0,0,0,&FF,&FF
  50  CLS
  60  DIM X(2),H(2),P(2)
  70  W=0:G=0:T=0
  80  FOR I=0 TO 2000:NEXT I:CLS
  90  P(2)=32768
 100  IF G=3 THEN 1010
 110  D=0:C=0:S=0:Y=0
 120  FOR A=1 TO 40
 130  PROCXY(33727+A,224)
 140  NEXT A
 150  FOR A=1 TO 3
 160  FOR B=1 TO 7
 170  PROCXY(33437+A+40*B,224)
 180  NEXT B
 190  NEXT A
 200  FOR A=1 TO 30
 210  PROCXY(33527+A,225)
 220  NEXT A
 230  IF C=1 THEN 260
 240  C=1:H(1)=RND(16)-1
 250  X(1)=0
 260  IF Y=1 THEN 530
 270  VDU 30:IF H(1)<2 THEN
      PRINT"DANGER-TOO LOW":VDU 30
 280  IF H(1)<0 THEN 930
 290  X(1)=X(1)+1
 300  IF X(1)>36 THEN 950
 310  FOR A=1 TO 5
 320  PROCXY(P(1)-1+A,32)
 330  NEXT A
 340  PROCXY(P(1)-40,32)
 350  P(1)=32768+X(1)+40*(16-H(1))
 360  PROCXY(P(1),224)
 370  PROCXY(P(1)+1,224)
 380  IF D=1 THEN PROCXY(P(1)+2,225):GOTO 400
 390  PROCXY(P(1)+2,224)
 400  PROCXY(P(1)+3,224)
 410  PROCXY(P(1)+4,226):PROCXY(P(1)-40,224)
 420  IF D=1 THEN 530
 430  A$=INKEY$(0)
 440  *FX 15,0
 450  FOR B=1 TO 30:NEXT B
 460  IF A$="" THEN 270
 470  IF A$="D" THEN H(1)=H(1)-1:GOTO 270
 480  IF A$="B" THEN 520
 490  IF H(1)>15 THEN A$="L"
 500  IF A$="U" THEN H(1)=H(1)+1
 510  GOTO 270
 520  F=H(1):D=1:X(2)=X(1)
 530  S=S+1
 540  H(2)=INT(F-S*S*0.5)
 550  X(2)=X(2)+1
 560  IF X(2)>29 THEN 660
 570  PROCXY(P(2),32)
 580  P(2)=32768+X(2)+40*(16-H(2))
 590  IF F<2 THEN 870
 600  PROCXY (P(2),224)
 610  IF H(2)<-1 THEN 630
 620  GOTO 260
 630  F=INT(3*F/4+0.5)
 640  S=-INT(SQR((F+2)*2)+0.5)
 650  GOTO 260
 660  G=G+1
 670  PROCXY(P(2),32)
 680  FOR A=1 TO 5
 690  PROCXY(P(1)-1+A,32)
 700  NEXT A
 710  PROCXY(P(1)-40,32)
 720  IF H(2)>0 THEN 80
 730  REM**DAM HIT
 740  FOR A=1 TO 3
 750  FOR B=1 TO 3
 760  PROCXY(33437+40*B+A,32)
 770  NEXT B
 780  NEXT A
 790  PROCXY(33599,32)
 800  PROCXY(33600,32)
 810  PROCXY(33639,32)
 820  FOR A=1 TO 40
 830  PROCXY(33527+A,225)
 840  NEXT A
 850  T=T+1
 860  GOTO 80
 870  FOR A=1 TO 5
 880  PROCXY(P(2)+40*A,224)
 890  FOR B=1 TO 15:NEXT B
 900  NEXT A
 910  PROCXY(P(2)+200,32)
 920  Y=1
 930  PRINT TAB(5,30);"TOO LATE - YOU
      HAVE CRASHED!";
 940  W=W+1
 950  FOR A=1 TO 5
 960  PROCXY(P(1)-1+A,32)
 970  NEXT A
 980  PROCXY(P(1)-40,32)
 990  G=G+1
1000  GOTO 80
1010  CLS:PRINT:PRINT"YOU HAVE LOST ";W;
      " AIRCRAFT."
1020  PRINT"YOU HAVE DESTROYED ";T;" DAMS."
1030  PRINT"THEREFORE YOUR SCORE IS ";T-W
1040  INPUT"DO YOU WANT ANOTHER GAME",B$
1050  IF LEFT$(B$,1)="Y" THEN RUN
1060  END
1070  DEFPROCXY(A,C)
1080  Y=INT((A-32768)/40)
1090  X=A-32768-40*Y
1100  PRINT TAB(X,Y);CHR$(C);
1110  ENDPROC
```

Converted from a program by Seamus Dunn

LEAPFROG

All you have to do is to swap over the coins from either side. Sounds simple?



Leapfrog is a computer implementation of a board game more usually played in pubs with coins. In its usual form there are six coins and seven spots, as shown in Fig. 1.

The purpose of the game is to interchange the two sets of coins and the rules are very simple.

1. The coins on the left can move only to the right and the coins on the right can move only to the left.
2. There are two possible types of move:
   (a) to an adjoining empty spot, or
   (b) over another coin to an empty spot.

Of course, the number of spots can be any odd number so that, for example, there might be four coins on each side and a spot in the middle making nine spots altogether. This program allows you to choose between three, five, seven and nine spots.

Jump To It

There are two distinct advantages to playing the game on a computer. In the first place, becoming clear about just what is permitted in the game is made easier because, while it will obviously allow you to make wrong or foolish moves, the computer will not allow you to make illegal or incorrect ones. This means that the machine forces you to learn the rule structure of the game very quickly allowing you to concentrate on strategy alone.

The second advantage is that it is possible for the computer to keep a record of all the moves made in any particular game. At the end of the game this record can be assessed and studied. In this way, wrong moves can be detected, winning patterns become clear and the structure of the game emerges.

Once you've played Leapfrog for a while, you will notice that patterns of movement begin to emerge and you may begin to wonder about how the game might be symbolised and its structure analysed. When that stage is reached, the educational and, in this case, mathematical aspects of the game become more interesting.

GAME ANALYSIS

The simplest version of the game is shown in Fig. 2.

A few quick trails will show that successful strategy is very simple and can be coded as LRL (or RLR). This means:
First move the L from 1 to 2.
Then move the R for 3 to 1.
Then move the L from 2 to 3.
The next version of the game is shown in Fig. 3.

In this case the successful strategy (if you start by moving 2) is coded as L RR LL RR L. This means:
First move an L piece.
Then move two R pieces.
Then move two L pieces.
Then move two R pieces.
Finally move an L piece.
Notice the pattern of the number moves is 12221. The strategies for versions of the game with three, five and seven spots are as follows:

LRL
L RR LL RR L
L RR LLL RRR LLL RR L

Turning these into number patterns as before:

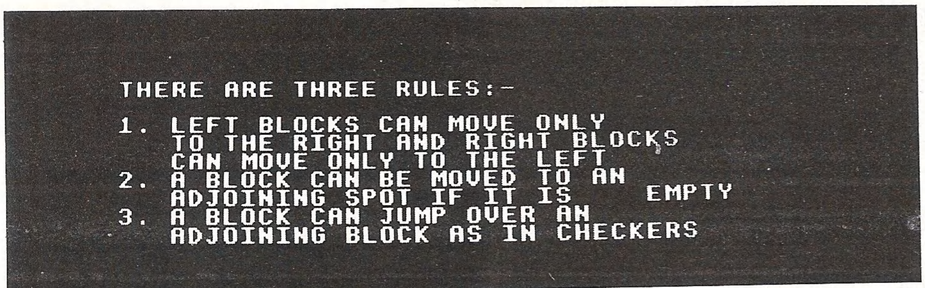
111
12221
1233321

we would guess that the nine spot version would become:

123444321

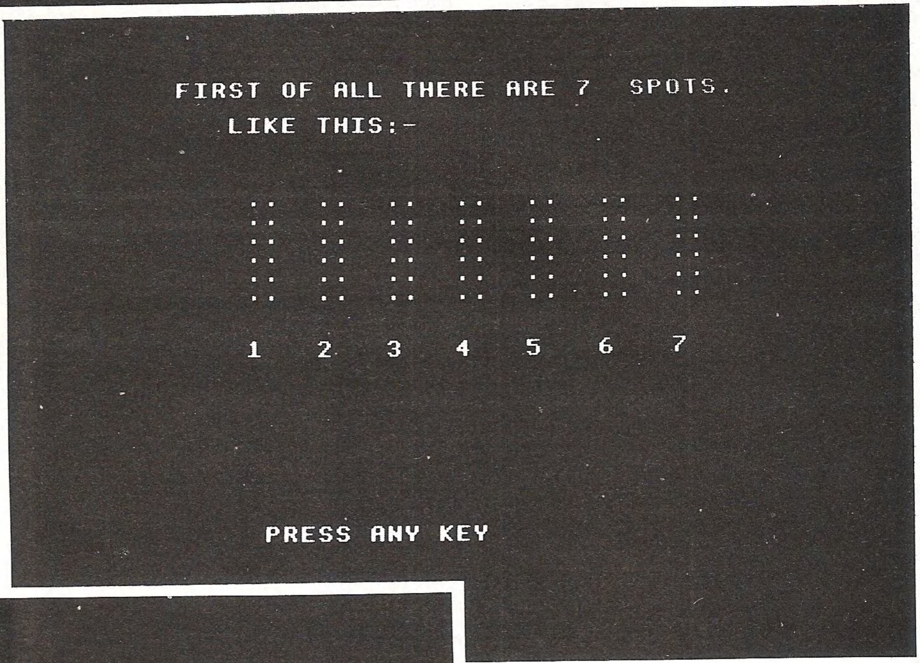
This signifies that you begin by moving the L piece, then two R pieces, then three L pieces and so on. You can try this out on the nine spot version on the computer and it does actually work.

At this stage the pattern is so simple and obvious that it is quite easy to succeed at any version of the game. But its analysis is not quite complete. The general case can be symbolised as the case with 2N ▶



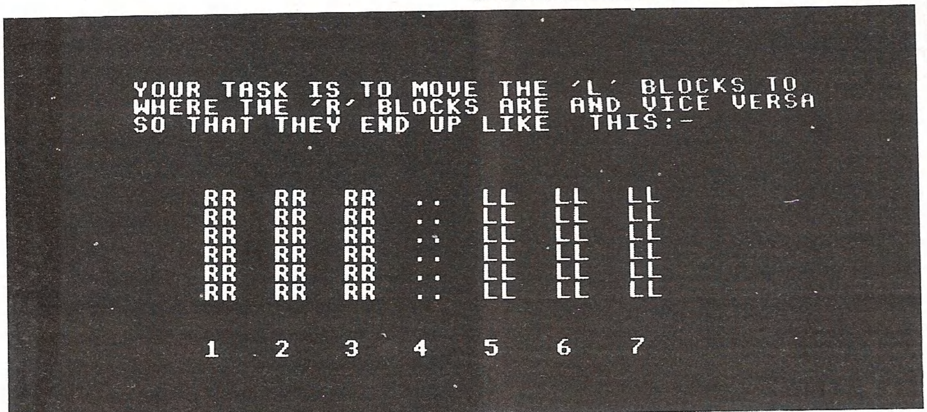
```
THERE ARE THREE RULES:-

1. LEFT BLOCKS CAN MOVE ONLY
   TO THE RIGHT AND RIGHT BLOCKS
   CAN MOVE ONLY TO THE LEFT
2. A BLOCK CAN BE MOVED TO AN
   ADJOINING SPOT IF IT IS     EMPTY
3. A BLOCK CAN JUMP OVER AN
   ADJOINING BLOCK AS IN CHECKERS





         PRESS ANY KEY
```

Left: The game rules as displayed by the initial stages of the program.



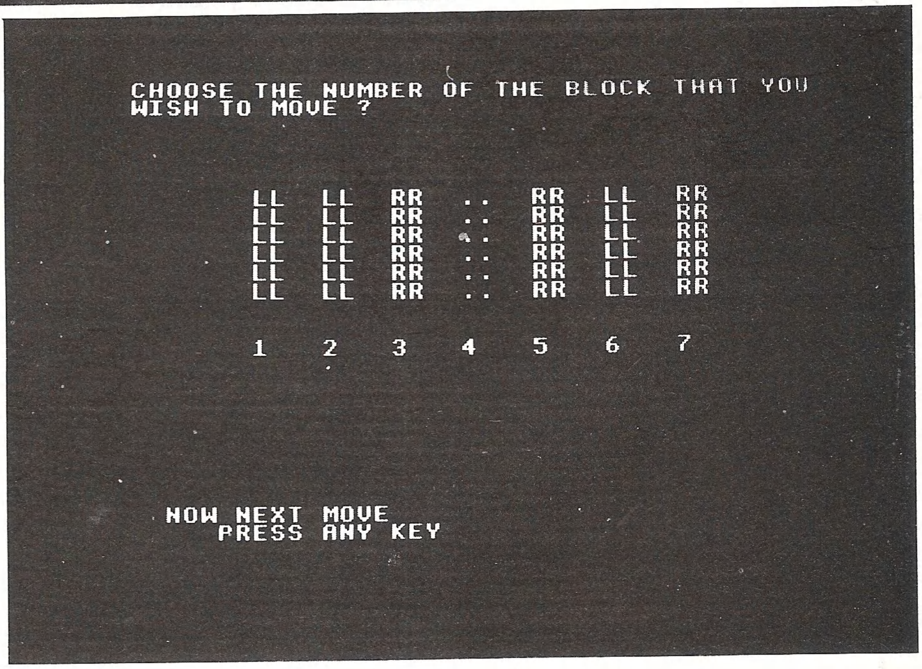
```
FIRST OF ALL THERE ARE 7  SPOTS.
  LIKE THIS:-


    ::   ::   ::   ::   ::   ::   ::
    ::   ::   ::   ::   ::   ::   ::
    ::   ::   ::   ::   ::   ::   ::
    ::   ::   ::   ::   ::   ::   ::
    ::   ::   ::   ::   ::   ::   ::

    1    2    3    4    5    6    7




     PRESS ANY KEY
```

Right: The program demonstrates a seven-spot game, this is the initial starting position . . .



```
YOUR TASK IS TO MOVE THE 'L' BLOCKS TO
WHERE THE 'R' BLOCKS ARE AND VICE VERSA
SO THAT THEY END UP LIKE   THIS:-


    RR   RR   RR   ::   LL   LL   LL
    RR   RR   RR   ::   LL   LL   LL
    RR   RR   RR   ::   LL   LL   LL
    RR   RR   RR   ::   LL   LL   LL
    RR   RR   RR   ::   LL   LL   LL
    RR   RR   RR   ::   LL   LL   LL

    1    2    3    4    5    6    7



     PRESS ANY KEY
```

Left: . . .and this is what the game is supposed to look like when you finish.



```
CHOOSE THE NUMBER OF THE BLOCK THAT YOU
WISH TO MOVE ?


    LL   LL   RR   ::   RR   LL   RR
    LL   LL   RR   ::   RR   LL   RR
    LL   LL   RR   ::   RR   LL   RR
    LL   LL   RR   ::   RR   LL   RR
    LL   LL   RR   ::   RR   LL   RR
    LL   LL   RR   ::   RR   LL   RR

    1    2    3    4    5    6    7




  NOW NEXT MOVE
     PRESS ANY KEY
```

Right: A seven-spot game in progress. The player has obviously not been reading the program notes as he is about to get somewhat stuck!

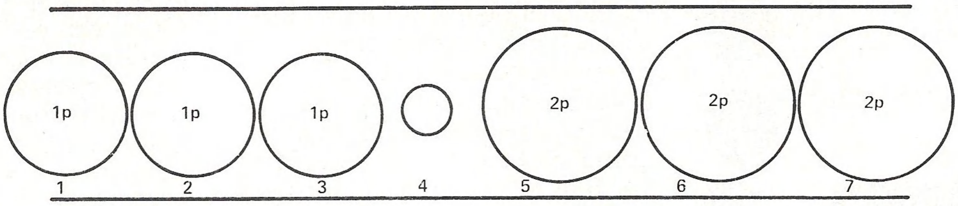


Fig. 1. The traditional game layout using coins.

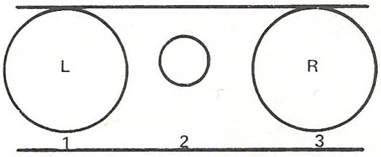


Fig. 2. The simplest possible game with the coins replaced by Left and Right markers.

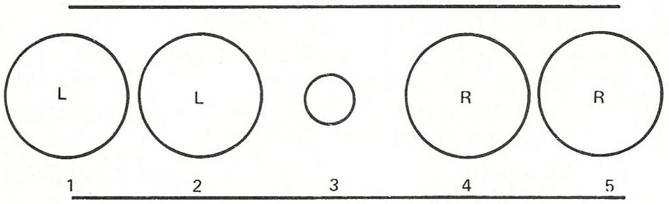


Fig. 3. The basic game as far as strategy development is concerned. If you can solve this one you can solve them all.

+ 1 spots. That is, N spots on the left, N spots on the right and one spot in the middle. So the pattern of movements can be shown as:

123. . .NNN. . .321

That is, one L move, two R moves, three L moves continuing until N moves are necessary. This number of moves (ie N) occurs three times and then the number begins to reduce by one each time until one move only is necessary. At this stage the game is over.

The final question may well be 'what is the total number of moves necessary at each level of the game' and a mathematical relationship can be used to answer this. In the general case, the total number of moves is:

$$(1+2+3+\ldots N) + N + (N+\ldots+3+2+1)$$

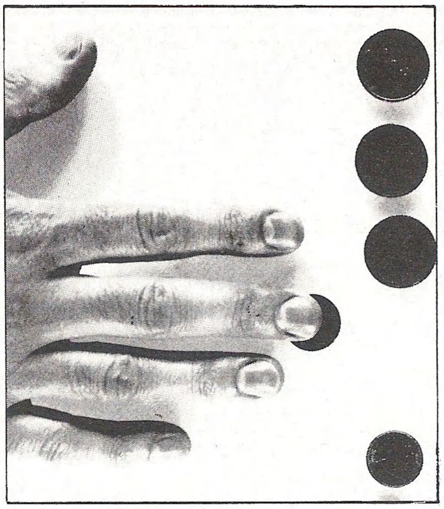
That is, $2(N)+N=N(N+2)$.

TECHNICAL DETAILS

Leapfrog is converted from PET BASIC and graphics. The only difficult part of this conversion ▶

PROGRAM STRUCTURE

Statement	Function	Action
Lines 20-120	Initial Display	Displays title and instructions if these are required.
Lines 130-190	Create Display	Sets up a display consisting of seven spots.
Lines 200-590	Example	Demonstrates play with seven spots.
Lines 600-680	Rules	Displays rules.
Lines 690-950	Select Game	Player asked to choose the numbers of spots. The appropriate number are then displayed.
Lines 960-1350	Move	Instructions requested from player about which spot is to be moved where. The legality of the move is checked. A move to the right is controlled in lines 1140-1240 and a move to the left is controlled in lines 1250-1350.
Lines 1360-1470	Error Message	If a disallowed move is attempted, an appropriate message is displayed.
Lines 1480-1660	Solution	Checks to see if correct solution has been obtained. If so, the computer prints congratulations and asks if details of solution are required. Offers another game.
Lines 1670-1760	Spots	Displays spots on the screen and numbers them.
Lines 1770-1840	Blank Screen	Blanks out messages from the screen.
Lines 1850-1870	Get Key	Initiates program when any key is pressed.
Lines 1880-1920	Address	Converts PET screen addresses to x,y co-ordinates.



is taken care of by PROCXY which converts PET screen addresses to x,y co-ordinates for the BBC Micro's PRINT TAB(X,Y);. No special graphics characters are used and so none are set up using the VDU 23 command.

You can usually take a PET BASIC conversion further by substituting BBC BASIC procedures which do the same thing. For example, if you look at lines 1830-1840 you will see a method of clearing part of the screen which results from directly translating a PET BASIC routine into BBC BASIC. If you analyse what is going on you should be able to see that all the routine does is to print a number of blanks at a particular position. This can be carried out more easily in BBC BASIC by PRINT TAB(X,Y); SPC(N);.

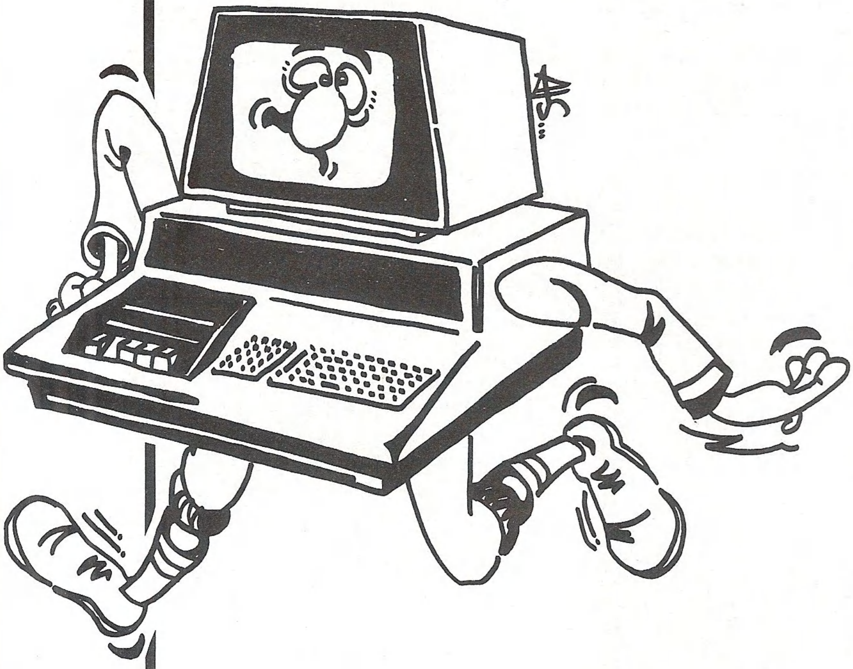
If you look at lines 1770-1790 you will find a routine clearing part of the screen, starting at the top, that uses just this method. The routine was originally a direct conversion from PET BASIC just like lines 1830-1840 but the neater BBC BASIC was substituted after the entire program was debugged and working.

How far you take a conversion is up to you. If the converted routine carries out its task fast enough for you then I would advise leaving well alone!

HOW TO PLAY

Once you've typed in the program it actually demonstrates a game being played. When it is your turn to play, you are asked to choose whether you want 3,5,7 or 9 spots displayed. The game is, of course, easier the fewer of the spots you try to handle. To make a move, you type in the number of the piece you wish to move and press the Return key. Then, simply type in the number of the space you wish to move it to.

The computer actually prompts you with appropriate questions and is quick to tell you if you try to make an illegal move. The computer will congratulate you if you succeed in swapping the pieces but if you fail you can get stuck so that no move is possible. If you wish to quit at any point, all you need do is type '99'. At the end of each game you are given the chance to study the moves you made and are asked if you want to play again.



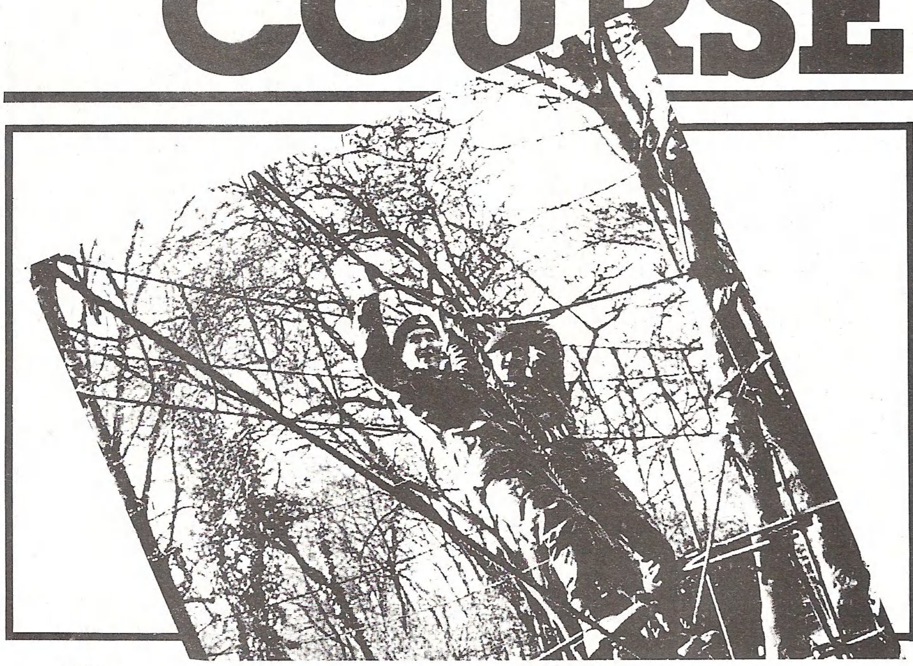
```
10   MODE 4
20   A$="---LEAPFROG---":R$=""
30   DIM A(10),B(10),SP(10)
40   CLS:PRINT:PRINT TAB((40-LEN(A$))/2); A$
50   PRINT:INPUT "DO YOU WANT
     INSTRUCTIONS.",B$
 60  IF LEFT$(B$,1)="Y" THEN 90
 70  IF LEFT$(B$,1)="N" THEN 690
 80  PRINT "TRY AGAIN":GOTO 40
 90  PRINT:PRINT "YOU CAN HAVE 3, 5, 7
     OR 9 POINTS IN"
100  PRINT "EACH GAME.  WE WILL
     DEMONSTRATE WITH 7"
110  PRINT "AND THEN YOU CAN CHOOSE."
120  GOSUB 1850:CLS
130  SP1=33052:REM**SCREEN POSITION 1
140  REM**SET UP SEVEN POINTS
150  FOR C=1 TO 7:SP(C)=SP1+4*(C-1):NEXT
160  FOR C=1 TO 3:A(C)=1:B(C)=2:NEXT:
     REM**  LEFT HAND POINTS
170  A(4)=0:B(4)=0:REM**MIDDLE POINT
180  FOR C=5 TO 7:A(C)=2:B(C)=1:NEXT:
     REM**   RIGHT HAND POINTS
190  GOSUB 1800
200  VDU 30:PRINT:PRINT "FIRST OF ALL
     THERE ARE 7  SPOTS."
210  PRINT:PRINT "   LIKE THIS:-"
220  S=46
230  FOR C=1 TO 7
240  P1=SP(C)
250  GOSUB 1670:REM**SETS UP 7 POINTS
260  GOSUB 1740:REM**NUMBERS POINTS
270  NEXT C
280  GOSUB 1850:GOSUB 1830:GOSUB 1800
290  VDU 30:PRINT:PRINT "THERE ARE 3  'L'
     BLOCKS":PRINT "ON THE LEFT LIKE THIS:"
300  S=76
310  FOR C=1 TO 3
320  P1=SP(C)
330  GOSUB 1670
340  NEXT
350  GOSUB 1850
360  GOSUB 1830:GOSUB 1800
370  VDU 30:PRINT:PRINT "THERE ARE 3
     'R'  BLOCKS ON"
380  PRINT "THE RIGHT....LIKE THIS:"
390  S=82
```

```
400  FOR C=5 TO 7
410  P1=SP(C)
420  GOSUB 1670
430  NEXT
440  GOSUB 1850:GOSUB 1830:GOSUB 1800
450  VDU 30:PRINT:PRINT "YOUR TASK IS
     TO MOVE THE 'L' BLOCKS TO"
460  PRINT "WHERE THE 'R' BLOCKS ARE
     AND VICE VERSA"
470  PRINT "SO THAT THEY END UP LIKE  THIS:-"
480  GOSUB 1850:GOSUB 1830
490  S=82
500  FOR C=1 TO 3
510  P1=SP(C)
520  GOSUB 1670
530  NEXT C
540  S=76
550  FOR C=5 TO 7
560  P1=SP(C)
570  GOSUB 1670
580  NEXT C
590  GOSUB 1850:GOSUB 1830
600  CLS:PRINT:PRINT "THERE ARE THREE RULES:-"
610  PRINT:PRINT "1. LEFT BLOCKS CAN
     MOVE ONLY"
620  PRINT SPC(3);"TO THE RIGHT AND
     RIGHT BLOCKS"
630  PRINT SPC(3);"CAN MOVE ONLY TO THE LEFT"
640  PRINT "2. A BLOCK CAN BE MOVED TO
     AN"
650  PRINT SPC(3);"ADJOINING SPOT IF
     IT IS  EMPTY"
660  PRINT "3. A BLOCK CAN JUMP OVER AN"
670  PRINT SPC(3);"ADJOINING BLOCK AS
     IN CHECKERS"
680  GOSUB 1850:GOSUB 1830
690  CLS:PRINT:INPUT "HOW MANY SPOTS
     DO YOU WANT (3,5,7,9)",Z
700  SP1=33048:REM**SCREEN POSITION 1
710  REM** SET UP Z POINTS
720  ZL=(Z-1)/2:ZM=(Z+1)/2:ZR=(Z+3)/2
730  FOR C=1 TO Z:SP(C)=SP1+4*C-2*Z+17:
     NEXT C
740  FOR C=1 TO ZL:A(C)=1:B(C)=2:NEXT
     C:REM**LEFT HAND POINTS
750  A(ZM)=0:B(ZM)=0:REM**MIDDLE POINTS
760  FOR C=ZR TO Z:REM**RIGHT HAND POINTS
770  A(C)=2:B(C)=1
780  NEXT C
790  CLS:PRINT
800  S=76
810  FOR C=1 TO ZL
820  P1=SP(C)
830  GOSUB 1670
840  GOSUB 1740
850  NEXT C
860  S=46
870  P1=SP(ZM)
880  GOSUB 1670
890  GOSUB 1740
900  S=82
910  FOR C=ZR TO Z
920  P1=SP(C)
930  GOSUB 1670
940  GOSUB 1740
950  NEXT C
960  GOSUB 1800
970  VDU 30:PRINT:PRINT "IF AT ANY
     TIME YOU WISH TO STOP PRESS 99"
980  GOSUB 1850:GOSUB 1830
990  GOSUB 1800
1000 VDU 30:PRINT:PRINT "CHOOSE THE
     NUMBER OF THE BLOCK THAT YOU"
1010 INPUT "WISH TO MOVE ",CS:
     REM**CURRENT SPOT
1020 IF CS>Z OR CS<0  THEN PRINT"NO
     SUCH SPOT ":GOTO 1000
1030 IF CS=99 THEN 1600:REM**END
1040 IF A(CS)=0 THEN 1360:REM**SPOT EMPTY
1050 GOSUB 1800
1060 VDU 30:PRINT:PRINT "CHOOSE THE
     NUMBER OF THE SPOT TO"
1070 INPUT "WHICH YOU WISH TO MOVE ", NS:
     REM**NEW SPOT  MOVE ?1
1080 IF NS>Z OR NS<0 THEN PRINT "NO
     SUCH SPOT":GOTO 1060
1090 IF NS=99 THEN 1600:REM**END
1100 IF ABS(CS-NS)>2 THEN 1390:
     REM**MORE THAN TWO APART
1110 IF A(NS)<>0 THEN 1420:REM**SPOT NOT EMPTY
1120 IF A(CS)=2 THEN 1250:REM**RIGHT TO LEFT
1130 IF A(CS)=1 THEN 1140:REM**LEFT TO RIGHT
1140 REM**MOVE LEFT TO RIGHT
1150 IF NS<=CS THEN 1450:REM**WRONG WAY
1160 R$=R$+"L"
1170 S=46:P1=SP(CS)
1180 GOSUB 1670
1190 S=76:P1=SP(NS)
1200 GOSUB 1670
1210 A(CS)=0:A(NS)=1
1220 PRINT TAB(2,24);"NOW NEXT MOVE"
1230 GOSUB 1800
1240 GOTO 1480:REM**CHECKS IF COMPLETE
1250 GOSUB 1670
1260 IF NS>CS THEN 1450:REM**WRONG WAY
1270 R$=R$+"R"
1280 S=46:P1=SP(CS)
1290 GOSUB 1670
1300 S=82:P1=SP(NS)
1310 GOSUB 1670
1320 A(CS)=0:A(NS)=2
1330 PRINT TAB(2,24);"NOW NEXT MOVE"
1340 GOSUB 1800
1350 GOTO 1480:REM**CHECKS IF COMPLETE
1360 REM**SPOT EMPTY
1370 PRINT "THERE IS NO BLOCK ON THIS SPOT"
1380 GOSUB 1850:GOSUB 1830:GOSUB 1770:
     GOTO 1000
1390 REM**MORE THAN 2 SPOTS APART
1400 PRINT "THE SPOTS ARE TOO FAR APART"
1410 GOSUB 1850:GOSUB 1830:GOSUB 1770:
     GOTO 1000
1420 REM**SPOT NOT EMPTY
1430 PRINT "THIS SPOT IS NOT EMPTY.  TRY AGAIN"
1440 GOSUB 1850:GOSUB 1830:GOSUB 1770:
     GOTO 1000
1450 REM**WRONG DIRECTION
1460 PRINT TAB(0,3);"YOU CANNOT MOVE
     IN THAT  DIRECTION, TRY AGAIN"
1470 GOSUB 1850:GOSUB 1830:GOSUB 1770:
     GOTO 1000
1480 REM**CHECKS IF COMPLETE
1490 GOSUB 1770
1500 FLAG=0
1510 FOR C=1 TO Z
1520 IF A(C)<>B(C) THEN FLAG=1
1530 NEXT C
1540 IF FLAG=1 THEN C=7:GOTO 990
1550 REM**COMPLETED
1560 GOSUB 1800
1570 PRINT TAB(0,24);"CONGRATULATIONS"
1580 PRINT "YOU HAVE SUCCEEDED"
1590 GOTO 1600
1600 INPUT "DO YOU WISH TO SEE YOUR
     SOLUTION ",Z$
1610 IF LEFT$(Z$,1)="Y" THEN 1630
1620 IF LEFT$(Z$,1)="N" THEN 1640
1630 PRINT "YOUR MOVES= ";R$
1640 INPUT"ANOTHER GAME Y/N",Z$
1650 IF LEFT$(Z$,1)="Y" THEN RUN
1660 END
1670 REM**MAKES SPOTS
1680 FOR C1=0 TO 200 STEP 40
1690 FOR C2=P1 TO P1+1
1700 PROCXY(C1+C2,S)
1710 NEXT C2
1720 NEXT C1
1730 RETURN
1740 REM**NUMBERS POINTS
1750 PROCXY(C1+C2+78,C+48)
1760 RETURN
1770 REM**CLEARS TOP OF SCREEN, LARGE
1780 PRINT TAB(0,0);SPC(240);
1790 RETURN
1800 REM**CLEARS TOP OF SCREEN, SMALL
1810 PRINT TAB(0,0);SPC(160)
1820 RETURN
1830 REM**CLEARS BOTTOM OF SCREEN
1840 FOR C=33688 TO 33708:PROCXY(C,32):
     NEXTC: RETURN
1850 PRINT TAB(5,25);"PRESS ANY KEY";
1860 Z$=GET$
1870 RETURN
1880 DEFPROCXY(A,C)
1890 Y=INT((A-32768)/40)
1900 X=A-32768-40*Y
1910 PRINT TAB(X,Y);CHR$(C)
1920 ENDPROC
```

Based on an idea by Mark Michalowski and G Lowry

OBSTACLE COURSE

It's a race against the clock to get through the moving maze!



Although a short program to type in, this game of skill is addictive — you have been warned!

The computer draws a colourful (yellow on a red background) maze which changes constantly throughout the game — as you head for a gap you can suddenly find your way blocked although equally, if you've the patience to wait, your path should eventually clear.

The time you spend in the maze is displayed throughout the game and when you've completed the course, the computer will confirm the time taken and tell you how many moves you made.

TECHNICAL DETAILS

The VDU's foreground colour has been set to yellow in line 40 and the background colour to red in line 50. Although there is no difficulty in playing the game as it stands on a black and white monitor or TV set, you may prefer to delete these lines to obtain a crisp white on black effect.

The standard methods of dealing with PET graphics and PEEK/POKE are used, ie PROCXY and FNS (for more details on these methods see the notes on Surround). The only special character created is 244 which is a stippled square.

HOW TO PLAY

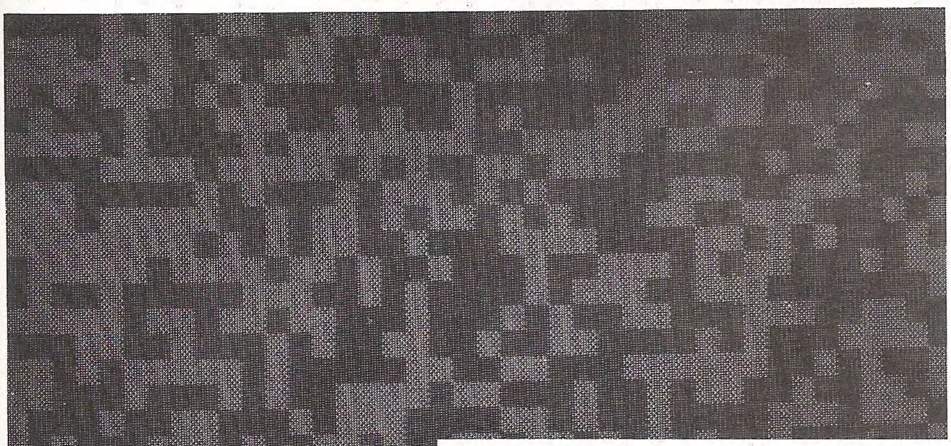
The object of the game is to move the asterisk, using the cursor control arrows at the top left of the keyboard, from the bottom left-hand corner of the screen to the '$' in the top right-hand corner. There are six levels of difficulty for the game (these levels determine the density of the maze): level one is fairly easy, level three is pretty difficult and level six is virtually impossible!

SCOPE FOR IMPROVEMENT

Try altering the message printed out according to how long the player took to complete the course. Thus the computer would give congratulations after a quick completion and rebukes if a longer time was needed.

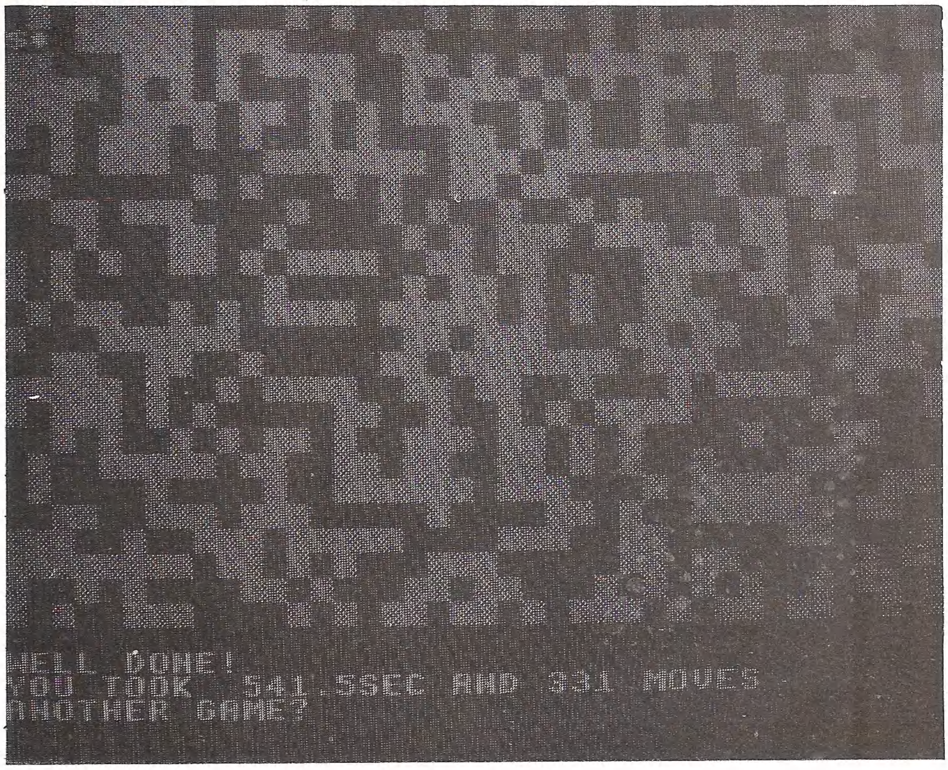
PROGRAM STRUCTURE

Statement	Function	Action
Lines 10-50	Set Up	Selects graphics mode, defines CHR$(224) as a stippled square and sets foreground and background colours. *FX 4,1 allows the arrow keys to be used.
Lines 60-90	Instructions	Choice of skill level for playing game.
Lines 100-120	Print Maze	Creates initial obstacle course.
Lines 130-160	Print Time	Displays time at bottom left of screen.
Lines 170-230	Get Key	Selects appropriate movement routines according to which arrow key is pressed.
Lines 240-270	Move Logic	Checks to see if asterisk has arrived at the target ($), bumped into the maze wall or moved off the screen.
Lines 280-360	End Game	Prints congratulations and score. Resets to normal mode (MODE 7) and restores arrow keys to usual function.
Lines 370-510	Change Maze	Removes existing blockages and creates new ones to give a dynamic obstacle course.



```
170.86 S
```

Left: The maze generated by the program is a constantly moving one. The asterisk is your current position, the dollar symbol (top left-hand corner) is the position you must reach. Your progress is timed, the elapsed time is continuously displayed at the bottom of the screen.



```
WELL DONE!
YOU TOOK  541.5SEC AND 331 MOVES
ANOTHER GAME?
```

Right: You've made it! Not a particularly rapid escape though, it took over 500 seconds and 331 moves.

```
 10  VDU 23,224,&AA,&55,&AA,&55,&AA,&55,&AA,&55
 20  *FX 4,1
 30  MODE 4
 40  VDU 19,1,3,0,0,0
 50  VDU 19,0,1,0,0,0
 60  CLS:PRINT:PRINT"WHICH LEVEL - 1 TO 6
     (6 HARDEST)";:INPUT U
 70  IF U<1 OR U>6 THEN 60
 80  IF INT(U)<>U THEN 60
 90  CLS
100  FOR I=1 TO (U*100):PRINT
     TAB(RND(40)-1,RND(25)-1);CHR$(224);:NEXT I
110  A=32768:B=33767
120  PROCXY(A):PRINT TAB(X,Y+1);"$";:
     PROCXY(B): PRINT TAB(X,Y);"*";
130  TIME=0
140  GOSUB 160
150  GOTO 140
160  PRINT TAB(10,27);TIME/100;" S";
170  GOSUB 370
180  A$=INKEY$(0):IF A$="" THEN 160
190  IF ASC(A$)=&8A THEN A=40:GOTO 240
200  IF ASC(A$)=&88 THEN A=-1:GOTO 240
210  IF ASC(A$)=&89 THEN A=1:GOTO 240
220  IF ASC(A$)=&8B THEN A=-40:GOTO 240
230  GOTO 360
240  PROCXY(A+B):D=?FNS(X,Y,0):IF D=12
     THEN 280
250  IF D<>0 THEN 360
260  IF (B+A)>33767 OR (B+A)<32768 THEN 360
270  F=F+1:PROCXY(B):PRINT TAB(X,Y);" ";:
     B=B+A:PROCXY(B):PRINT TAB(X,Y);"*";:
     GOTO 360
280  PRINT TAB(0,26);"WELL DONE!"
290  PRINT TAB(0,27);"YOU TOOK  ";
300  PRINT TAB(15,27);"SEC AND ";F;" MOVES"
310  INPUT "ANOT!iER GAME",A$
320  IF LEFT$(A$,1)="Y" THEN GOTO 60
330  *FX 4,0
340  MODE 7
350  END
360  RETURN
370  REM
380  X=RND(40)-1:Y=RND(25)-1
390  D=?(FNS(X,Y,3))
400  IF D<>0 AND D<>85 THEN 420
410  PRINT TAB(X,Y);" ";
420  X=RND(40)-1:Y=RND(25)-1
430  D=?(FNS(X,Y,3))
440  IF D<>0 AND D<>85 THEN 460
450  PRINT TAB(X,Y);CHR$(224);
460  RETURN
470  DEFPROCXY(M)
480  Y=INT((M-32768)/40)
490  X=M-32768-40*Y
500  ENDPROC
510  DEF FNS(X,Y,N)=HIMEM+(Y*40+X)*8+N
```

BBC MICRO GAMES

THREE SUPERB GAMES
CASSETTES AVAILABLE **NOW**
FOR THE BBC MICRO (MODELS A AND B)

CASSETTE ONE:- Two games demonstrating the fantastic capabilits in sound and graphics of the BBC Micro.
(1) **STAR TREK** – The classic computer game. Try your hand at commanding the Star Ship Enterprise.
(2) **CANDY FLOSS** – A tremendous new game in which you test your business skills running a candy floss stall on Blackpool's Golden Mile, but watch the weather and the donkeys!

– ONLY £5.95 inc

CASSETTE TWO:- An exciting collection of games with music and graphics to keep the family amused for hours:- HANGMAN, KRYPTOGRAM, DICE, BEETLE, GRAND NATIONAL AND MUSIC.

– ONLY £3.95 inc

CASSETTE THREE:- For the arcade fanatics:- MUTANT INVASION – a brilliant new "SPACE INVADERS" type game. WARNING!!! very addictive.

– ONLY £5.95 inc

Deduct £1 for 2 Cassettes & £2 for 3 Cassettes
send cheque/P.O. to
Sinclair, Dept C.T.
55 Fitzroy Road, Bispham, Blackpool, Lancs.

BBC SOFTWARE

WORDWISE – THE WORD PROCESSOR for the BBC Micro – A professionally written program supplied in ROM, containing all the usual word processor facilities plus several unusual ones like a fully automatic word count. Extremely fast and very simple to use, a powerful piece of software. Supplied with a full manual for £65 + £1.50 p&p + VAT.

SNAKE – This highly addictive game for the model B makes full use of the amazing graphics and sound facilities of the BBC Micro, it will give hours of fun. Supplied on a quality cassette for £5.80 + VAT.

RAM KITS. Upgrade the model A to 32K RAM – supplied with full instructions for £29.90 + VAT.

CASH or royalties waiting for any interesting software for the BBC machine. The better the program the more we will pay.

Send an SAE at least 6" by 9" for latest details of our software and to get your name on our mailing list. Please add VAT to all prices. Mail order only.

COMPUTER CONCEPTS

Dept CT6
16 Wayside
Chipperfield
Herts WD4 9JJ
Tel. (09277) 62955

JOIN THE CLUB

Would you like to discover more about your BBC Micro from other users? Why not join a user group? The following are all user groups devoting themselves to the BBC Micro.

Computer Users Club

72 Sidmouth Road,
Welling,
Kent DA16 1DS
Tel: 01-304 3910
Secretary: Tony Latham

The club produces a monthly printout of software ideas for the BBC Micro, programs and advice on programming technique.

BEEBUG

The National Independent User Group For The BBC Micro
35 St Julians Road,
St Albans,
Herts
Contact: D E Graham or Sheridan Williams

BEEBUG runs a regular newsletter (10 issues a year) including program listings, hardware and software tips, reviews and advice, all exclusively devoted to the BBC Micro. Membership is available for £4.50 for six months or £8.50 for the full year's subscription.

LASERBUG

The London And South-East Region BBC Microcomputer User Group
4 Station Bridge,
Woodgrange Road,
Forest Gate,
London E7 0NF
Contact: Trevor Sharples

The monthly newsletter, also called LASERBUG, contains reviews of books, hardware and software, as well as a number of programs to use on the BBC Micro. Membership is £12 and includes 12 month's subscription to the newsletter.

Independent BBC Users' Group

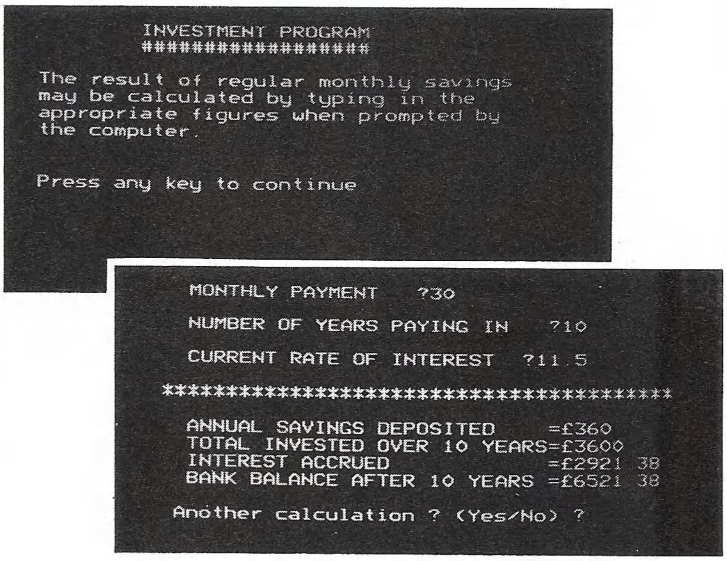
Dept. 1A,
44-46 Earls Court Road,
London W8 6EJ

This group has been in operation since October, 1981, and is run by Jeremy Ruston and Tim Hartnell who have written a book of programs for the BBC Micro published by the BBC. Priced at 85p, INTERFACE the Club's publications, covers news, reviews and programs for the BBC Micro.

Financial

Investment Checker — Having saved some money, this will help you work out what your profits would amount to should you invest a regular amount over a period of time.

1.5K



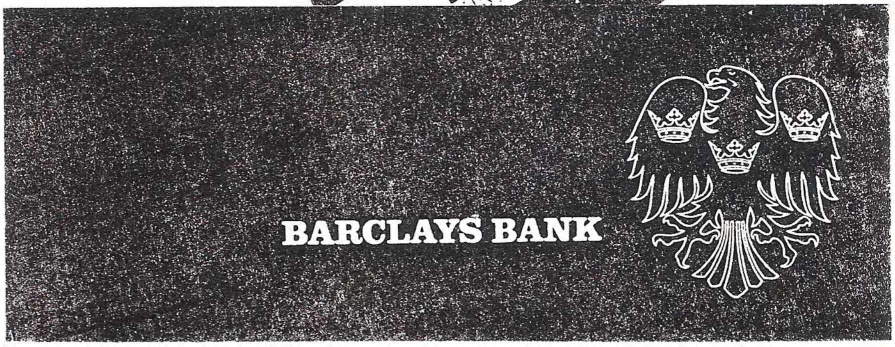
```
        INVESTMENT PROGRAM
        ###################

The result of regular monthly savings
may be calculated by typing in the
appropriate figures when prompted by
the computer.


Press any key to continue
```

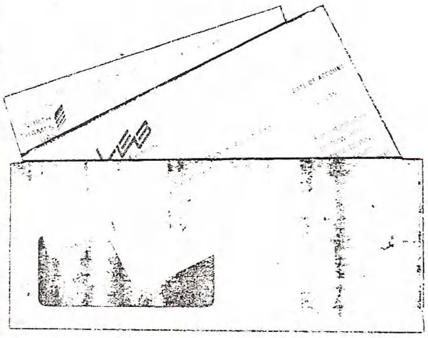
```
   MONTHLY PAYMENT    ?30

   NUMBER OF YEARS PAYING IN    ?10

   CURRENT RATE OF INTEREST  ?11.5

***************************************

   ANNUAL SAVINGS DEPOSITED     =£360
   TOTAL INVESTED OVER 10 YEARS=£3600
   INTEREST ACCRUED             =£2921.38
   BANK BALANCE AFTER 10 YEARS =£6521.38

  Another calculation ? (Yes/No) ?
```

Home Finance — Use this program to keep track of your finances. The computer will help you to budget by doing the sums and presenting you with all the up to date information you need to keep within your spending limits. The program will handle up to 20 accounts.

7.5K

BARCLAYCARD
VISA





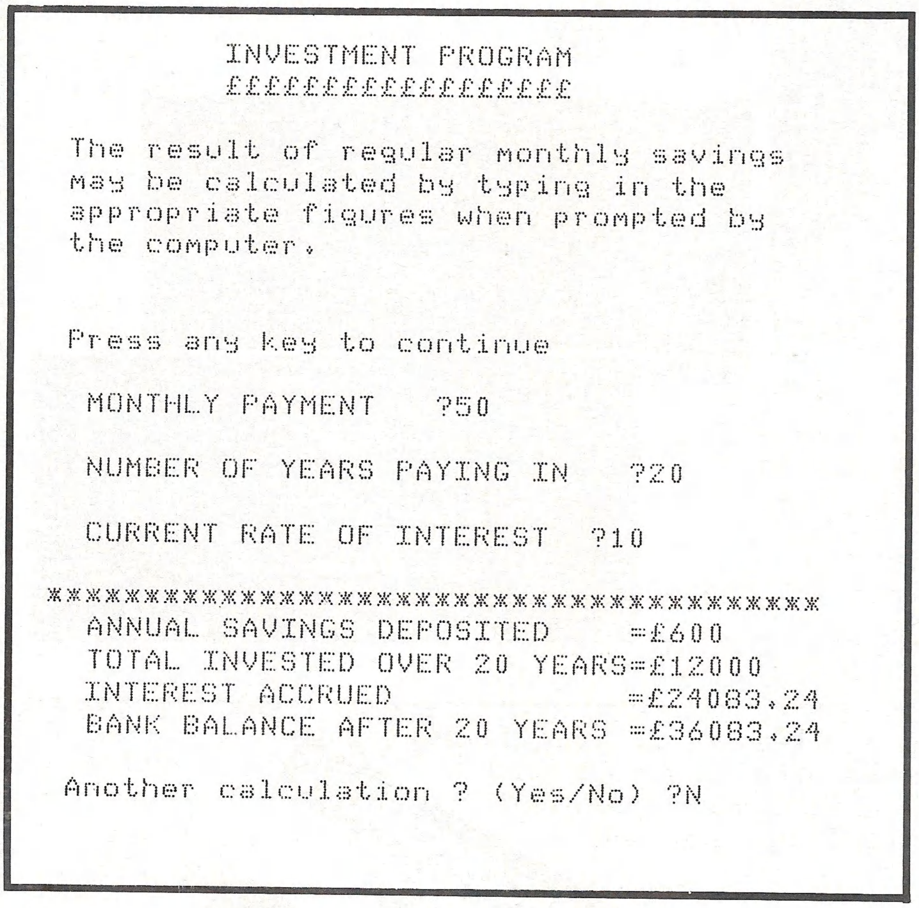
Meter Minder — This package provides you with a series of projections based on your meter readings . . . gas and electricity. Fuel consumptions in cost per week, month or quarter. Plan how to SAVE IT.

2.5K

Converted from a program by J Consadine

INVESTMENT CHECKER

Insuring you get the most from your money can be tricky these days!



```
          INVESTMENT PROGRAM
          ££££££££££££££££££

 The result of regular monthly savings
 may be calculated by typing in the
 appropriate figures when prompted by
 the computer.


 Press any key to continue

  MONTHLY PAYMENT    ?50

  NUMBER OF YEARS PAYING IN    ?20

  CURRENT RATE OF INTEREST  ?10

*****************************************
  ANNUAL SAVINGS DEPOSITED     =£600
  TOTAL INVESTED OVER 20 YEARS=£12000
  INTEREST ACCRUED             =£24083.24
  BANK BALANCE AFTER 20 YEARS  =£36083.24

 Another calculation ? (Yes/No) ?N
```

PROGRAM STRUCTURE

Statement	Function	Action
Lines 10-20	Set Up	Defines array and clears screen.
Lines 30-120	Instructions	Prints title and instructions.
Lines 130-160	Input	Requests input.
Lines 170-200	Preparation	Sets array to zero and prints header line.
Lines 210-350	Calculate & Print Results	Calculates and prints annual savings, calculates annual bank balance (lines 250-270), calculates and prints total invested, interest accrued and final bank balance.
Lines 360-380	End	Asks if another calculation is required. If so, requests input, otherwise ends.

Those of you lucky, or careful enough, to have any money left over when you've settled all your bills have a problem this program can help you with: the problem of deciding where you will get the best return for your money.

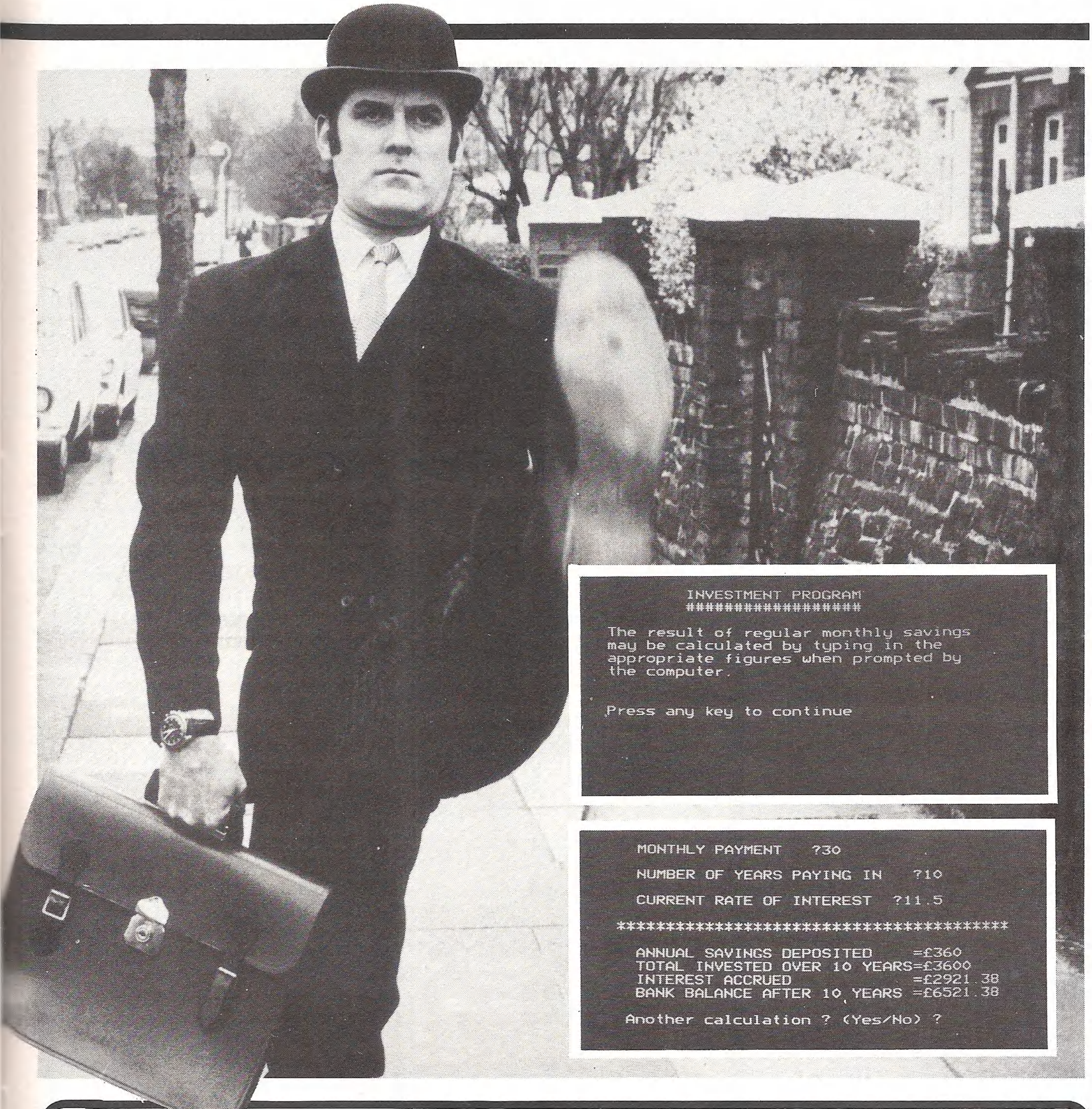
Is your best option to increase the payments on your Life Assurance policy so that when it matures you will receive extra bonus payments? Or is it best to save with a Savings Bank or Building Society? Making comparisons can be difficult if the insurance companies only give figures for the maturity value over twenty years.

This program presents the results of investing a regular sum at a known interest rate over any number of years. It thus projects the effect of saving any sum over a long period (without the tedious calculations for compound interest) and so enables comparisons to be made.

HOW TO USE THE PROGRAM

The program asks you to provide the computer with three figures: the amount of the monthly payment; the number of years you wish to project your savings over; and the current rate of interest.

It then displays the annual investment you deposit, the total you will have invested over the period specified, the interest accrued and the eventual bank balance. Note that the program will only work for complete years (you cannot ask it to calculate on the basis of, say, two and a half years).



```
        INVESTMENT PROGRAM
        ##################

 The result of regular monthly savings
 may be calculated by typing in the
 appropriate figures when prompted by
 the computer.


 Press any key to continue
```

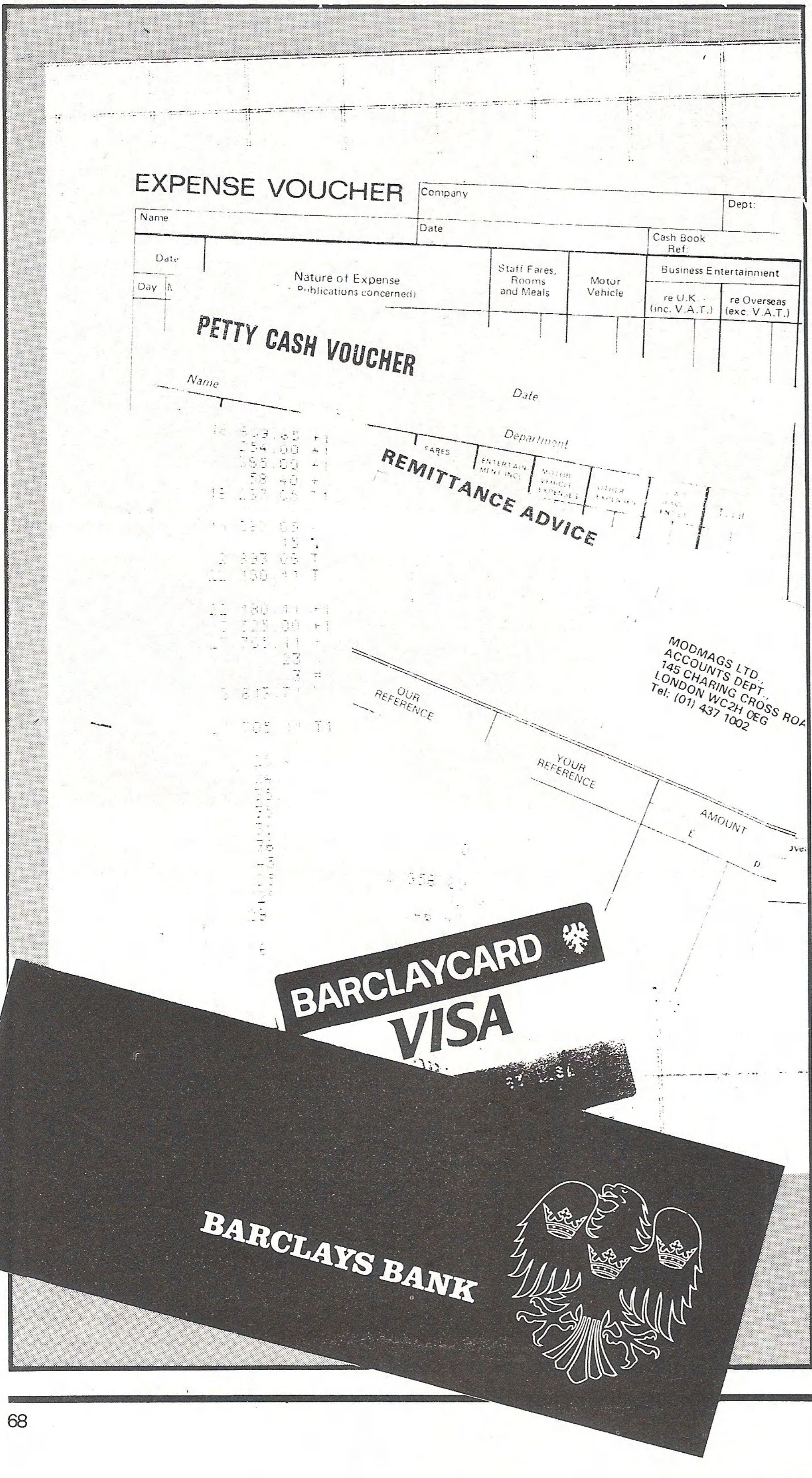
```
    MONTHLY PAYMENT    ?30

    NUMBER OF YEARS PAYING IN    ?10

    CURRENT RATE OF INTEREST  ?11.5

 ****************************************

    ANNUAL SAVINGS DEPOSITED     =£360
    TOTAL INVESTED OVER 10 YEARS=£3600
    INTEREST ACCRUED             =£2921.38
    BANK BALANCE AFTER 10 YEARS =£6521.38

   Another calculation ? (Yes/No) ?
```

```
 10  DIM B(100)
 20  CLS
 30  PRINT
 40  PRINT"          INVESTMENT PROGRAM"
 50  PRINT"          ££££££££££££££££££"
 60  PRINT
 70  PRINT" The result of regular
     monthly savings"
 80  PRINT" may be calculated by
     typing in the"
 90  PRINT" appropriate figures when
     prompted by"
100  PRINT" the computer."
110  PRINT:PRINT:PRINT" Press any key
     to continue"
120  A$=GET$
130  CLS
140  PRINT:INPUT"  MONTHLY PAYMENT    ",P
150  PRINT:INPUT"  NUMBER OF YEARS
     PAYING IN    ",Y
160  PRINT:INPUT"  CURRENT RATE OF
     INTEREST  ",R
170  PRINT
180  PRINT STRING$(40,"*");
190  PRINT
200  FOR N=0 TO Y:B(N)=0:NEXT:REM**ZERO ARRAY
210  S=12*P
220  PRINT"  ANNUAL SAVINGS DEPOSITED  ";
     TAB(30);"=`";S
230  A=S/2:REM**AVERAGE ANNUAL SAVINGS
240  REM**CALCULATE ANNUAL BANK BALANCE
250  FOR N=1 TO Y
260  B(N)=S+(A*R/100)+B(N-1)+(B(N-1)*R/100)
270  NEXT N:T=S*Y
280  PRINT"  TOTAL INVESTED OVER ";Y;
     " YEARS";TAB(30);"=`";T
290  PRINT"  INTEREST ACCRUED";
300  IA=(INT((B(Y)-T)*100))/100
310  PRINT TAB(30);"=`";IA
320  PRINT"  BANK BALANCE AFTER ";Y;" YEARS";
330  B(Y)=(INT(B(Y)*100))/100
340  PRINT TAB(30);"=`";B(Y)
350  PRINT
360  INPUT" Another calculation ?
     (Yes/No) ",A$
370  IF LEFT$(A$,1)="Y" THEN 130
380  END
```

Converted from a program by Elaine Douse.

HOME FINANCE

Use this program to keep track of your finances.


EXPENSE VOUCHER
Company
Dept:
Name
Date
Cash Book Ref:
Nature of Expense
Staff Fares, Rooms and Meals
Motor Vehicle
Business Entertainment
re U.K. (inc. V.A.T.)
re Overseas (exc. V.A.T.)
PETTY CASH VOUCHER
Name
Date
Department
REMITTANCE ADVICE
MODMAGS LTD.,
ACCOUNTS DEPT.,
145 CHARING CROSS ROA
LONDON WC2H 0EG
Tel: (01) 437 1002
OUR REFERENCE
YOUR REFERENCE
AMOUNT
BARCLAYCARD
VISA
BARCLAYS BANK

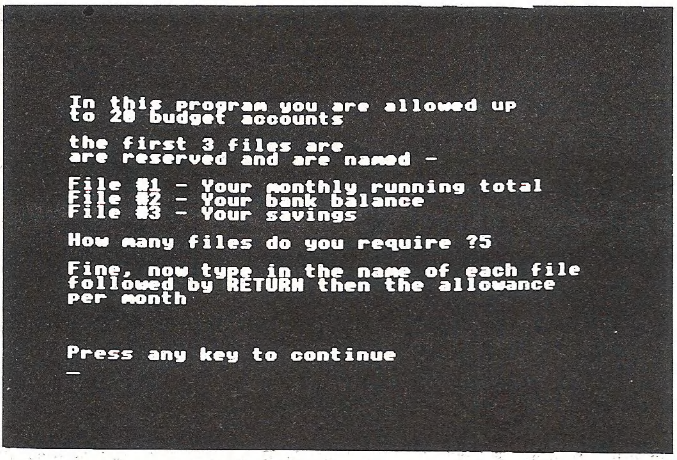
This program will help you to gain control of your spending. It helps you to budget by doing the sums and presenting you with the up to date information you need to keep within your spending limits.

The program handles up to 20 accounts, the first three of which it has allocated for special purposes, namely the monthly running total (which is the amount of money left to spend according to your budget), your bank balance and your savings. This leaves 17 un-allocated accounts. You can use as many of them as you need deciding what categories to allocate them to and what amount of money you think is reasonable to spend on each every month, for example, you might allocate an allowance of £20 a month to a gas account. Then every month, when you pay your salary in, the budgeted amount is added to your gas account and when you settle the gas bill, the amount you pay is deducted from the account.

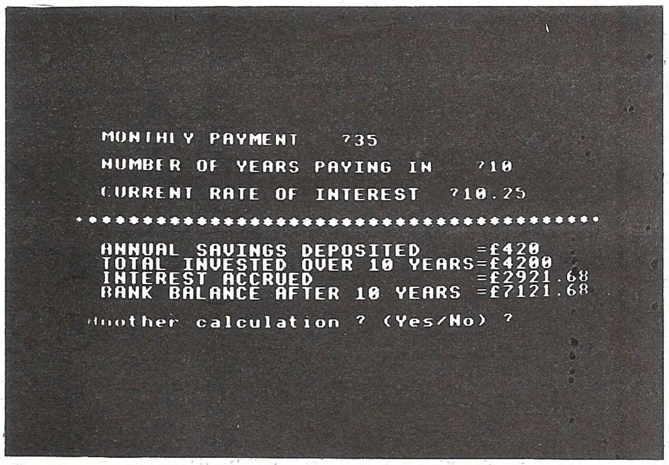
HOW TO USE THE PROGRAM

The first time you use the program, answer 'YES' to the first question 'is this a new program' and type in the budget names and amounts. Then save the file on tape with the correct date.

The second and subsequent times you use the program, answer 'NO' to the first question and load the file produced from the last run. Once a month you will have to input your salary; the rest of the time you must draw cheques from each of the accounts as you spend money



```
In this program you are allowed up
to 20 budget accounts

the first 3 files are
are reserved and are named -

File #1 - Your monthly running total
File #2 - Your bank balance
File #3 - Your savings

How many files do you require ?5

Fine, now type in the name of each file
followed by RETURN then the allowance
per month


Press any key to continue
_
```



```
MONTHLY PAYMENT     ?35

NUMBER OF YEARS PAYING IN     ?10

CURRENT RATE OF INTEREST   ?10.25

*************************************

ANNUAL SAVINGS DEPOSITED      =£420
TOTAL INVESTED OVER 10 YEARS=£4200
INTEREST ACCRUED              =£2921.68
BANK BALANCE AFTER 10 YEARS  =£7121.68

Another calculation ? (Yes/No) ?
```

PROGRAM STRUCTURE

Statement	Function	Action
Lines 10-60	Set Up	Selects MODE 4, defines characters used for graphs and sets up variables and arrays.
Lines 70-100	Start	Prints title. Asks user whether program is new (ie being used for a new budget).
Lines 110-280	Menu	Displays options for use of program and jumps to appropriate routine.
Lines 290-310	Input	General input routine.
Lines 320-480	Salary Input	Asks for amount of salary to be input and updates bank balance and savings.
Lines 490-730	Cheque Clearing	Routine to deduct money from accounts and from bank balance.
Lines 740-880	Display Accounts	Prints the number of its account and its title.
Lines 890-1060	Display Budgets	Prints the budget allowed for each account.
Lines 1070-1430	Salary Graph	Displays graph of salary input each month.
Lines 1440-1780	Capital Graph	Displays graph of savings each month.
Lines 1790-2020	Alter Accounts	Allows user to re-name accounts and to change amounts budgeted.
Lines 2030-2290	Set up Budgets	Initial allocation of files to accounts with allowances.
Lines 2300-2600	Save onto Tape	Saves data onto cassette tape. User asked for date and number of last cheque.
Lines 2610-2630	Y/N Input	General routine for 'Y' and 'N' input.
Lines 2640-2920	Load from Tape	Allows user to re-load data previously saved on tape.
Lines 2930-3080	Transfer	Allows user to transfer money between accounts.
Lines 3090-3130	Print	Routine to print names and amounts in each account.
Lines 3140-3180	Screen Address	Converts PET screen memory address to x,y co-ordinates. Prints character C.

on the budget item. If you become overdrawn on any account then you can use option 10 to transfer money into it.

As you use the program other categories of things which you spend money on might occur to you. If this happens then use option seven to create new budgets from unused ones. If you find an account is becoming regularly overdrawn, you can also use option seven to change the budget allowance.

Every time you draw a cheque on an account, the amount is subtracted from the account, your current bank balance and the monthly running total. By looking at the monthly running total you can see how much of this month's salary is unspent.

If you decide to save some money, the sum is added to the savings account and subtracted from your bank balance and monthly running total. To draw on your savings account use option 10 to transfer money to your bank account. You can display a graph of your total capital (ie savings plus bank account) using option six. After completing all your transactions, save the data on tape using option nine.

BARCLAYCARD
VISA
BARCLAYS BANK

▶

```
10   MODE 4
20   VDU23,224,&FF,&FF,&FF,&FF,&FF,&FF,&FF,&FF
30   VDU23,225,&AA,&55,&AA,&55,&AA,&55,&AA,&55
40   N1=0:N2=0:N3=0:CH=0
50   DIM B$(20),B(20),T(20),M1(12),S1(12)
60   CLS
70   PRINT:PRINT TAB(7);"BANK ACCOUNT -
     HOME FINANCE"
80   PRINT:PRINT:PRINT "Is this a new
     program (Yes/No)"
90   GOSUB 300:IF R$ ="Y" THEN 2030
100  GOSUB 2640
110  CLS:PRINT:PRINT TAB(17);"M E N U":PRINT
120  PRINT TAB(4);"1. SALARY INPUT,
     BUDGET UPDATE "
130  PRINT TAB(4);"2. CHEQUE CLEARING "
140  PRINT TAB(4);"3. DISPLAY ACCOUNTS "
150  PRINT TAB(4);"4. DISPLAY BUDGETS "
160  PRINT TAB(4);"5. GRAPHIC DISPLAY
     OF SALARY "
170  PRINT TAB(4);"6. GRAPHIC DISPLAY
     OF CAPITAL "
180  PRINT TAB(4);"7. ALTER BUDGET
     TITLE AND ALLOWANCE "
190  PRINT TAB(4);"8. INITIAL SET-UP "
200  PRINT TAB(4);"9. UPDATE TAPE FILE "
210  PRINT TAB(3);"10. INTER ACCOUNT TRANSFER"
220  PRINT
230  INPUT "Please type category required ",R
240  IF INT(R)<>R THEN 110
250  IF R<1 OR R>10 THEN 110
260  CLS
270  ON R GOTO 320,490,740,890,1070,
     1440.1790,2030,2300,2930
280  GOTO 110
290  PRINT:PRINT "Press any key to continue"
300  R$=GET$
310  RETURN
320  PRINT:PRINT "DATA WILL BE
     DESTROYED IF I CONTINUE"
330  PRINT
340  PRINT "Are you happy for me to
     proceed ?":GOSUB 300
350  IF R$<>"Y" THEN 110
360  M1(N2)=T(2)+T(3)
370  PRINT
380  INPUT "What is this months take
     home pay please ",T(1):S1(N2)=T(1)
390  REM IF YOU HAVE ANY MONTHLY STANDING
400  REM ORDERS INSERT T(1)=T(1)-amount
410  REM of standing order here
420  T(2)=T(2)+T(1)
430  PRINT:PRINT "After standing orders your"
440  PRINT "balance is ";T(1)
450  PRINT "Your current savings are ":
     PRINT T(3)
460  PRINT "Your bank balance stands at
     ":PRINT T(2)
470  GOSUB 290
480  FOR I=1 TO 20:T(I)=T(I)+B(I):NEXT:
     GOTO 110
490  CLS:PRINT
500  FOR I=1 TO 20:GOSUB 3090:PRINT:NEXT
510  INPUT "Which file do you require",R
520  IF R<1 OR R>20 THEN 510
530  CH=CH+1
540  PRINT "Cheque No. ";CH;", please
     type in amount";
550  INPUT R1
560  T(2)=T(2)-R1:T(1)=T(1)-R1
570  IF R<>3 THEN 600
580  T(3)=T(3)+R1
590  GOTO 610
600  T(R)=T(R)-R1
610  IF T(2)>0 THEN 630
620  PRINT "WARNING BANK BALANCE IS ";
     T(2);" IN THE RED"
630  IF T(1)>0 THEN 650
640  PRINT "WARNING MONTHLY BALANCE IS  ";
     T(1);" IN DEBT"
650  IF T(R)<0 THEN 670
660  PRINT "YOUR ";B$(R);" ACCOUNT
     STANDS AT ";T(R):GOTO 680
670  PRINT "WARNING !! ";B$(R);"
     ACCOUNT IS ";T(R);" IN THE RED"
680  PRINT "Have you more for this account?"
690  GOSUB 300
700  IF R$="Y" THEN 530
710  PRINT "Have you any more cheques
     to clear ?":GOSUB 300
720  IF R$="Y" THEN 490
730  GOTO 110
740  PRINT:FOR I=1 TO 20:GOSUB 3090
750  T=INT(T(I)*100)/100:A$=STR$(T):
     GOSUB 790
760  PRINT TAB(38-LEN(A$));A$
770  NEXT
780  GOSUB 290:GOTO 110
790  IF T<>0 THEN 810
800  A$="0.00":RETURN
810  IF INT(T)<>T THEN 830
820  A$=A$+".00":RETURN
830  IF INT(T*10)<>T*10 THEN 850
840  A$=A$+"0"
850  IF T<1 THEN 870
860  RETURN
870  A$="0"+RIGHT$(A$,LEN(A$)-1)
880  RETURN
890  PRINT:A=0
900  FOR I=1 TO 20
910  A=A+B(I)
920  GOSUB 3090
930  T=INT(ABS(B(I))*100)/100
940  A$=STR$(T)
950  GOSUB 790
960  PRINT TAB(38-LEN(A$));A$
970  NEXT
980  PRINT:PRINT "Your budget per month is";
990  REM IF YOU HAVE ANY MONTHY STANDING
1000 REM ORDERS INSERT A=A+standing order
     amount here
1010 T=INT(ABS(A)*100)/100
1020 A$=STR$(T)
1030 GOSUB 790
1040 PRINT TAB(38-LEN(A$));A$
1050 GOSUB 290
1060 GOTO 110
1070 PRINT
1080 PRINT TAB(20); "SALARIES"
1090 PRINT "900 1"
1100 PRINT "    1"
1110 PRINT "    1"
1120 PRINT "    1"
1130 PRINT "800 1"
1140 PRINT "    1"
1150 PRINT "    1"
1160 PRINT "    1"
1170 PRINT "700 1"
1180 PRINT "    1"
1190 PRINT "    1"
1200 PRINT "    1"
1210 PRINT "600 1"
1220 PRINT "    1"
1230 PRINT "    1"
1240 PRINT "    1"
1250 PRINT "500
     1___________________________________ "
1260 PRINT TAB(6);"J  F  M  A  M  J  J
     A  S  O  N  D"
1270 P=33491
1280 FOR I=1 TO 12
1290 B=0
1300 P=P+3
1310 FOR J=500 TO S3(I)STEP 25
1320 IF S3(I)<=J THEN 1360
1330 IF I=N2 THEN 1390
1340 PROCXY(P+B,224)
1350 B=B-40
1360 NEXT J
1370 NEXT I
1380 GOTO 1410
1390 PROCXY(P+B,225)
1400 GOTO 1350
1410 PRINT TAB(1,28);
1420 GOSUB 290
1430 GOTO 110
1440 PRINT:PRINT TAB(15);"SAVINGS"
1450 PRINT "2000 1"
1460 PRINT "     1"
1470 PRINT "     1"
1480 PRINT "     1"
1490 PRINT "     1"
1500 PRINT "1500 1"
1510 PRINT "     1"
1520 PRINT "     1"
1530 PRINT "     1"
1540 PRINT "     1"
1550 PRINT "1000 1"
1560 PRINT "     1"
1570 PRINT "     1"
```

```
1580  PRINT "      1"
1590  PRINT "      1"
1600  PRINT " 500 1"
1610  PRINT TAB(5);STRING$(24,"-")
1620  PRINT "      J F M A M J J A S O N D"
1630  P=33412
1640  FOR I=1 TO 12
2200  FOR I=4 TO 20:B$(I)="SPARE":NEXT
2210  B$(1)="MONTHLY TOTAL"
2220  B$(2)="BANK BALANCE "
2230  B$(3)="SAVINGS      "
2240  YPOS=20
2250  PRINT:PRINT "FILE #";TAB(10);
      "NAME";TAB(20);"AMOUNT":PRINT
2260  FOR I=4 TO R
2270  PRINT TAB(2,VPOS);I;TAB(10,VPOS);:
      INPUT B$(I):PRINT " ";TAB(20,VPOS-1);:
      INPUT B(I)
2280  NEXT
2290  GOTO 110
2300  PRINT: PRINT "I AM ABOUT TO RECORD
      ALL YOUR DATA ONTO"
2310  Z=2
2320  PRINT "THE DATA TAPE"
2330  PRINT "FIRST YOU MUST VERIFY THE
      FOLLOWING"
2340  PRINT "TODAY'S DATE IS ";N1;"_";N2;"_";N3
2350  GOSUB 2610
2360  IF R$="Y" THEN GOTO 2380
1650  B=0
1660  P=P+2
1670  FOR J=500 TO M4(I) STEP 100
1680  IF M4(I)<=J THEN 1720
1690  IF I=N2 THEN 1750
1700  PROCXY(P+B,224)
1710  B=B-40
1720  NEXT J
1730  NEXT I
1740  GOTO 1770
1750  PROCXY(P+B,225)
1760  GOTO 1710
1770  PRINT TAB(0,28);:GOSUB 290
1780  GOTO 110
1790  PRINT
1800  FOR I=1 TO 20
1810  GOSUB 3090
1820  PRINT
1830  NEXT
1840  PRINT "Which account do you wish
      to change"
1850  INPUT R
1860  IF R<4 OR R>20 THEN 1790
1870  IF INT(R)<>R THEN 1790
1880  Z$=B$(R):Z=B(R)
1890  INPUT "What is the new name ",B$(R)
1900  INPUT "How much is its allowance ",
      B(R)
1910  PRINT "CHECK"
1920  PRINT "Old ";Z$;" is now ";B$(R)
1930  PRINT "And the allowance of ";Z;
      " is now ";B(R)
1940  PRINT "IS THIS CORRECT ?"
1950  GOSUB 300
1960  IF R$="Y" THEN 1980
1970  B$(R)=Z$:B(R)=Z:GOTO 1790
1980  PRINT "Do you wish to lose this budget's"
1990  PRINT "running total?"
2000  GOSUB 300
2010  IF R$="Y" THEN T(R)=0
2020  GOTO 110
2030  CLS:PRINT:PRINT "In this program
      you are allowed up"
2040  PRINT "to 20 budget accounts "
2050  PRINT
2060  PRINT "the first 3 files are"
2070  PRINT "are reserved and are named - "
2080  PRINT
2090  PRINT "File #1 - Your monthly
      running total "
2100  PRINT "File #2 - Your bank balance"
2110  PRINT "File #3 - Your savings "
2120  PRINT:PRINT"How many files do you
      require ";
2130  INPUT R
2140  IF R<4 OR R>20 OR R<>INT(R) THEN 2030
2150  PRINT:PRINT "Fine, now type in the
      name of each file"
2160  PRINT "followed by RETURN then the
      allowance"
2170  PRINT "per month"
```

```
2180  PRINT
2190  PRINT:GOSUB 290
2370  GOSUB 2880
2380  F=F+1
2390  PRINT"THIS WILL BE FILE NO, "F""
2400  PRINT "THE LAST CHEQUE WAS ";CH
2410  GOSUB 2610
2420  IF R$="Y" THEN 2450
2430  PRINT "A NEW BOOK !"
2440  INPUT "WHAT WAS THE LAST CHEQUE NO.",CH
2450  Z$="BANK"+STR$(F)
2460  FILE=OPENOUT(Z$)
2470  FOR I=1 TO 20:PRINT #FILE,B(I):NEXT
2480  FOR I=1 TO 20:PRINT #FILE,T(I):NEXT
2490  FOR I=1 TO 20:PRINT #FILE,B$(I):NEXT
2500  FOR I=1 TO 12:PRINT #FILE,M5(I):NEXT
2510  FOR I=1 TO 12:PRINT #FILE,S5(I):NEXT
2520  PRINT #FILE,CH
2530  PRINT #FILE,N1
2540  PRINT #FILE,N2
2550  PRINT #FILE,N3
2560  CLOSE#FILE
2570  Z=Z-1
2580  IF Z=1 THEN  2460
2590  PRINT "I HAVE FINISHED _ BYE FOR NOW"
2600  END
2610  PRINT "ANSWER Y OR N"
2620  GOSUB 300
2630  RETURN
2640  PRINT:PRINT "Please insert bank
      data tape and"
2650  PRINT "and fully rewind"
2660  GOSUB 290
2670  INPUT "Which file number do you
      require?",F
2680  Z$ ="BANK" + STR$(F)
2690  FILE=OPENIN(Z$)
2700  PRINT Z$;" FOUND"
2710  FOR I=1 TO 20:INPUT #FILE,B(I)
2720  NEXT I
2730  FOR I=1 TO 20:INPUT #FILE,T(I)
2740  NEXT I
2750  FOR I=1 TO 20:INPUT #FILE,B$(I)
2760  NEXT I
2770  FOR I=1 TO 12:INPUT #FILE,M6(I)
2780  NEXT I
2790  FOR I=1 TO 12:INPUT #FILE,S6(I)
2800  NEXT I
2810  INPUT #FILE,CH
2820  INPUT #FILE,N1
2830  INPUT #FILE,N2
2840  INPUT #FILE,N3
2850  CLOSE #FILE
2860  PRINT "This file was recorded on ";
      N1;"_";N2;'_";N3
2870  PRINT "The last cheque was ";CH
2880  PRINT "Please type in today's date"
2890  INPUT "Day",N1
2900  INPUT "Month",N2
2910  INPUT "Year",N3
2920  RETURN
2930  PRINT
2940  FOR I=1 TO 20
2950  GOSUB 3090:PRINT
2960  NEXT
2970  PRINT "From which file are you
      transferring"
2980  INPUT "money", I
2990  IF I<0 OR I>20 OR I<>INT(I) THEN 2930
3000  PRINT "Into which file are you
      transferring"
3010  INPUT "money", A
3020  IF A=I OR A<1 OR A>20 OR A<>INT(A)
      THEN 3000
3030  INPUT "And the amount please", R
3040  T(I)=T(I)-R:T(A)=T(A)+R
3050  PRINT "Have you any more transfers?"
3060  GOSUB 2610
3070  IF R$="Y" THEN 2930
3080  GOTO 110
3090  IF I>9 THEN 3110
3100  PRINT TAB(5);I;:GOTO 3120
3110  PRINT TAB(4);I;
3120  PRINT TAB(7);B$(I);
3130  RETURN
3140  DEFPROCXY(A,C)
3150  Y=INT((A-32768)/40)
3160  X=A-32768-40*Y
3170  PRINT TAB(X,Y);CHR$(C);
3180  ENDPROC
```

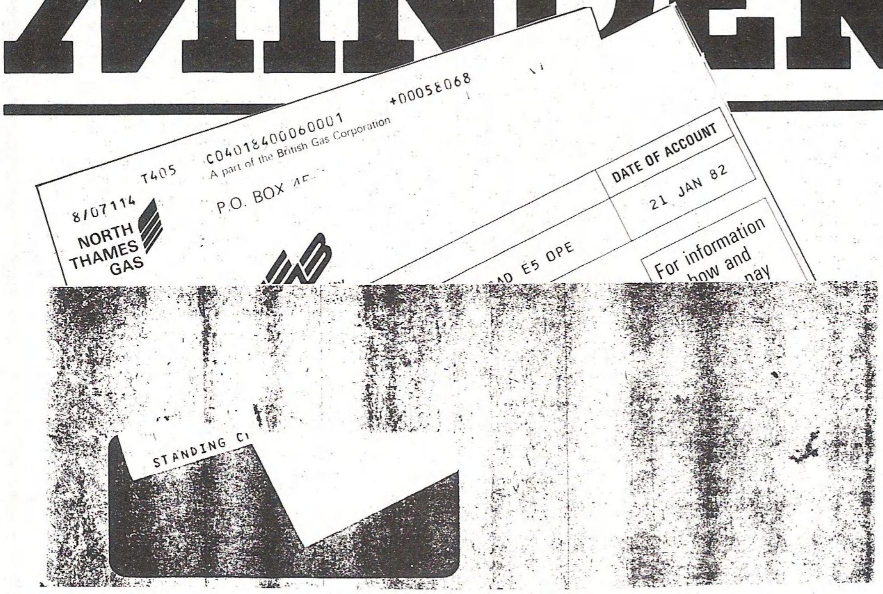
Converted from a program by John Hiscott

METER MINDER

Plan your energy consumption with your BBC Micro.

With the high cost of domestic energy there is always the worry about how much money to put aside for the next fuel bill. But, never fear, this program will give you an estimate to budget on.

The computer will provide you with projections based on your meter readings. It will tell you how much your fuel consumption is costing per week, per month and per quarter — so you can plan either to save money or to save energy in order to be able to foot the dreaded quarterly bill.



PROGRAM STRUCTURE

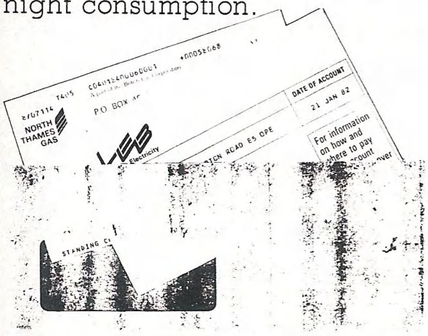
Statement	Function	Action
Lines 10-30	Set Up	Clears screen, defines print format and initialises variables.
Lines 40-60	Title	Displays title.
Lines 70-100	Fuel	Offers choice of projections for gas or electricity.
Lines 110-160	Gas	Asks for details of gas cost and consumption.
Lines 170-240	Electricity	Asks for details of electricity costs.
Lines 250-260	Quarterly Charge	Asks for amount of standing charge per quarter.
Lines 270-330	Readings	Choice of entering data for a single reading or a range of them.
Lines 340-360	Electricity Consumption	Asks for details of consumption of electricity if a single tariff applies.
Lines 370-390	Electricity Consumption	Asks for details of consumption of electricity if a two-part tariff applies.
Lines 400-480	Headings	Prints headings for table to display results.
Lines 490-640	Calculations	Computes results for both gas and electricity.
Lines 650-710	Results	Prints results in table; lines 680-700 dealing with those results relating to a two-part tariff for electricity.

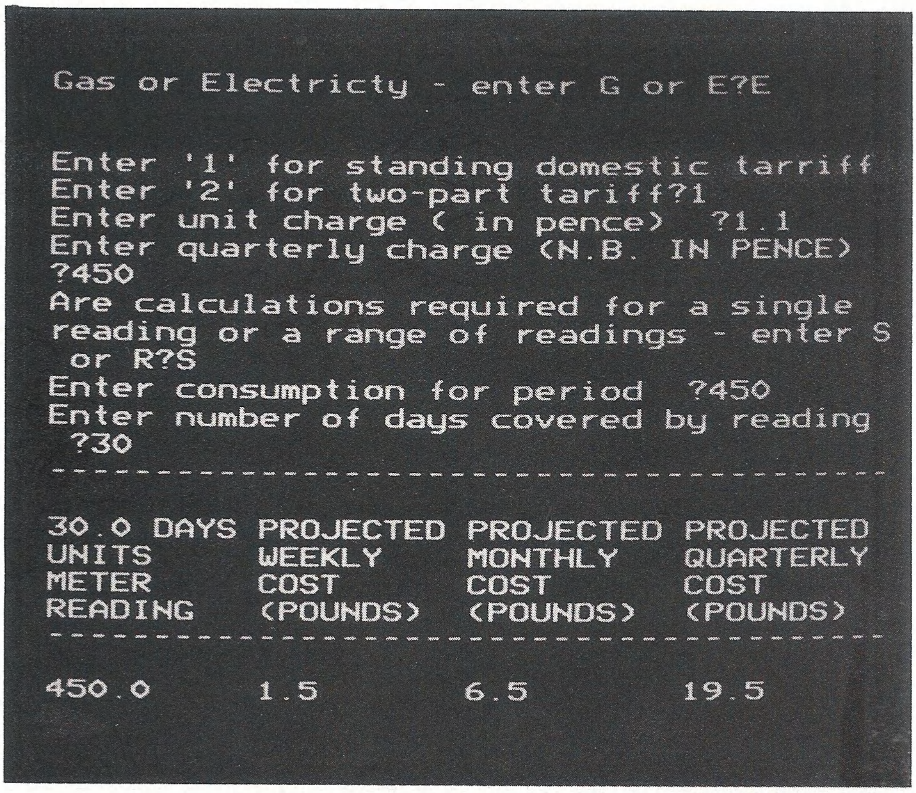
HOW TO USE THE PROGRAM

You will need to collect some information before you can get this program to make any projections for you. All the details you need about costs will be available from your last fuel bills and you can consult your meter to discover your current (!) consumption. The program requires your consumption in units over a known number of days and will either print a single projection for weekly/monthly/quarterly costs or will print a table of how much you will pay weekly/ monthly/quarterly for a range of consumptions over the known period. For example, if you've used 100 units in two days then you can either discover how much this will cost you per week/ month/quarter or you can ask for a table starting at 50 units per two days going up to 150 units per two days in steps of ten.

This table can also be used to discover how much your final bill can be altered by reducing your consumption. You can then make statements to the rest of your family like — 'If you save 10 units a day we will be XX pounds better off at the end of a quarter!'.

If you want to estimate a two-part tariff electricity bill then you can only ask for a single estimate because a table would be of no use in deciding the effects of transferring consumption from day to night rate. If you want to do this you will have to run the program with different values of day and night consumption.





```
Gas or Electricty - enter G or E?E

Enter '1' for standing domestic tarriff
Enter '2' for two-part tariff?1
Enter unit charge ( in pence)  ?1.1
Enter quarterly charge (N.B. IN PENCE)
?450
Are calculations required for a single
reading or a range of readings - enter S
 or R?S
Enter consumption for period  ?450
Enter number of days covered by reading
 ?30
----------------------------------------

30.0 DAYS PROJECTED PROJECTED PROJECTED
UNITS     WEEKLY    MONTHLY   QUARTERLY
METER     COST      COST      COST
READING   (POUNDS)  (POUNDS)  (POUNDS)
----------------------------------------

450.0     1.5       6.5       19.5
```

```
 10  CLS
 20  @1=&00020106
 30  B=0:P=0:E=0:L=1:R=0:I=0:C$="S":Y=0:U=0
 40  PRINT:PRINT "   DOMESTIC FUEL
     COSTING PROGRAM"
 50  PRINT "
     =============================="
 60  PRINT:PRINT
 70  INPUT "Gas or Electricty - enter
     G or E",A$
 80  A$=LEFT$(A$,1)
 90  IF A$="E" THEN GOTO 170
100  IF A$<>"G" THEN GOTO 70
110  PRINT:PRINT
120  INPUT "Enter first cost per therm
     (in pence)  ",A
130  INPUT "Enter number of therm
     charged at above rate  ",B
140  INPUT "Enter second cost per
     therm (in pence)  ",C
150  INPUT "Enter calorific value (BTU
     per Cu. Ft.)  ",X
160  GOTO 250
170  PRINT:PRINT
180  PRINT "Enter '1' for standing
     domestic tarriff"
190  INPUT "Enter '2' for two-part
     tariff",E
200  IF E=2 GOTO 230
210  INPUT "Enter unit charge (in pence)  ",A
220  GOTO 250
230  INPUT "Enter unit charge - day
     rate (in pence)  ",A
240  INPUT "Enter unit charge - night
     rate (in pence)  ",C
250  INPUT "Enter quarterly charge
     (N.B. IN PENCE)  ",I
260  IF E=2 GOTO 370
270  PRINT "Are calculations required
     for a single"
280  INPUT "reading or a range of
     readings - enter S or R",  C$
290  IF C$="S" THEN GOTO 340
300  PRINT:INPUT "Enter lowest reading
     of range",J
310  PRINT:INPUT "Enter highest
     reading of range",K
320  INPUT "Enter value of steps  ",L
330  GOTO 390
340  IF E=2 THEN GOTO 370
350  INPUT "Enter consumption for
     period  ",Y
360  C$="S":GOTO 390
370  INPUT "Enter day-rate consumption
     for period  ",U
380  INPUT "Enter night-rate
     consumption for period  ",V
390  INPUT "Enter number of days
     covered by reading  ",D
400  PRINT STRING$(40,"-")
410  PRINT TAB(0,VPOS);D;" DAYS";
420  PRINT TAB(10);"PROJECTED PROJECTED
     PROJECTED"
430  IF A$="G" THEN PRINT "CU FT";
440  IF A$="E" THEN PRINT "UNITS";
450  PRINT TAB(10);"WEEKLY    MONTHLY
     QUARTERLY"
460  PRINT "METER     COST      COST
       COST"
470  PRINT"READING   (POUNDS)  (POUNDS)
      (POUNDS)"
480  PRINT STRING$(40,"-")
490  IF C$="S" THEN M=Y:J=Y:K=Y
500  IF E=2 THEN M=Y:J=Y:K=Y
510  FOR M=J TO K STEP L
520  N=91/D
530  IF A$="E" THEN GOTO 570
540  O=M*X/1000*N
550  IF O>=B THEN P=O-B
560  IF O>=B THEN O=B
570  IF E=1 THEN O=M*N
580  IF E=2 THEN O=U*N:P=V*N
590  Q=O*A
600  IF E=1 THEN GOTO 620
610  R=P*C
620  S=Q+R+I
630  T=S/100
640  IF E=2 THEN GOTO 680
650  PRINT TAB(0,VPOS);M;TAB(10);T/13;
     TAB(20);T/3;TAB(30);T
660  NEXT M
670  END
680  PRINT TAB(0,VPOS);V+U;TAB(10);T/13;
     TAB(20);T/3;TAB(30);T
690  PRINT
700  PRINT "(COMBINED DAY PLUS NIGHT UNITS)"
710  END
```

Personal SOFTWARE

Personal Software is a new quarterly publication from the people who bring you Computing Today. To celebrate the launch of the BBC Microcomputer our first issue will consist of more than 20 programs covering Domestic, Financial, Educational, Games and Scientific areas.

All the programs are fully tested and documented and the listings have been produced directly from the BBC Micro to eliminate errors. As an additional service we are offering copies of the programs on tape through our CT Software organisation.

As well as featuring the best software from previous issues of Computing Today converted for the BBC Micro in order to show off its advanced features, the publication also includes a number of specially commissioned programs which reveal *even more* special functions.

If you own or have ordered a BBC Micro, or are just looking for a collection of Extended BASIC programs to convert to your system, then you need Personal Software: BBC Programs.

Personal Software will be on sale at your local newsagent from Friday 14th May at £1.95 or you can order directly from us at £7.80 per annum or £1.95 per copy. To ensure a single copy or a complete year's supply fill in the form below — you can even spread the load with your credit card.

SUBSCRIPTION ORDER FORM

Cut out and SEND TO :

Personal SOFTWARE

513 LONDON ROAD,
THORNTON HEATH,
SURREY,
CR4 6AR.

POST FREE SUBSCRIPTION

Please use BLOCK CAPITALS and include post codes.

Name (Mr/Mrs/Miss) ....
delete accordingly

Address ....

....

....

Signature ....

Date ....

82 Summer

Please commence my subscription to Personal Software with the .... issue.

SUBSCRIPTION RATES (tick ☐ as appropriate)

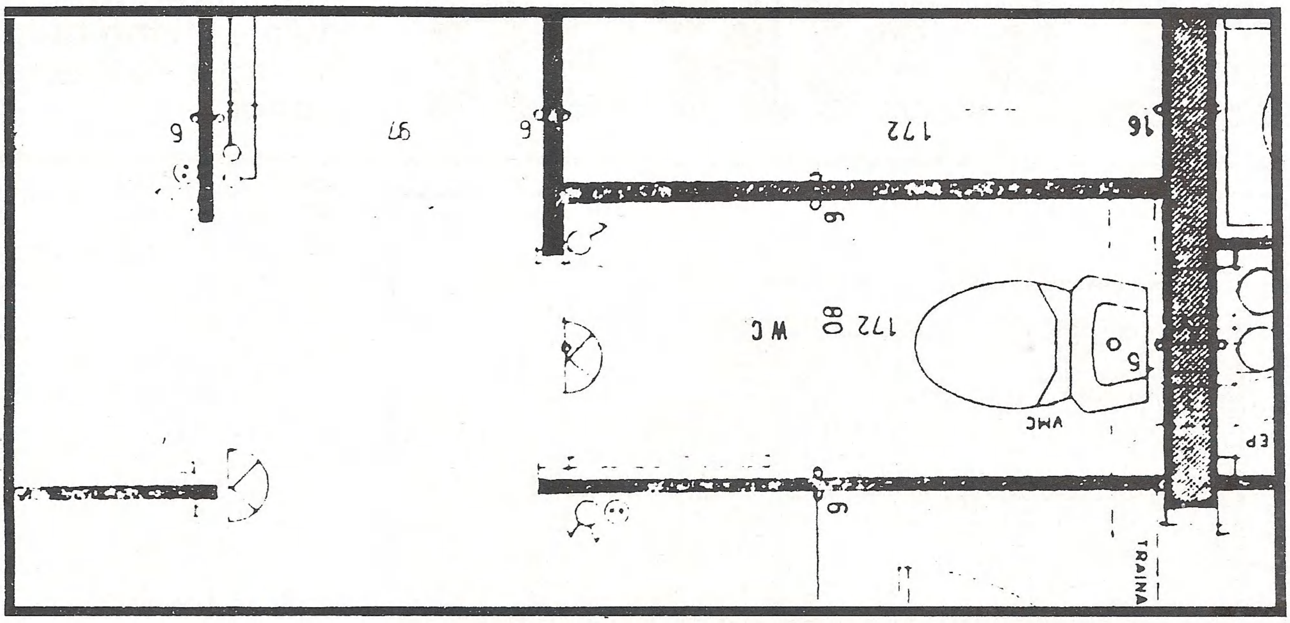
£7.80 for 4 issues UK ☐

£1.95 for a single copy of the .... issue ☐

I am enclosing my (delete as necessary) Cheque/Postal Order/International Money Order for £.........
(made payable to ASP Ltd)
OR
*Debit my Access/Barclaycard**
*(*delete as necessary)*

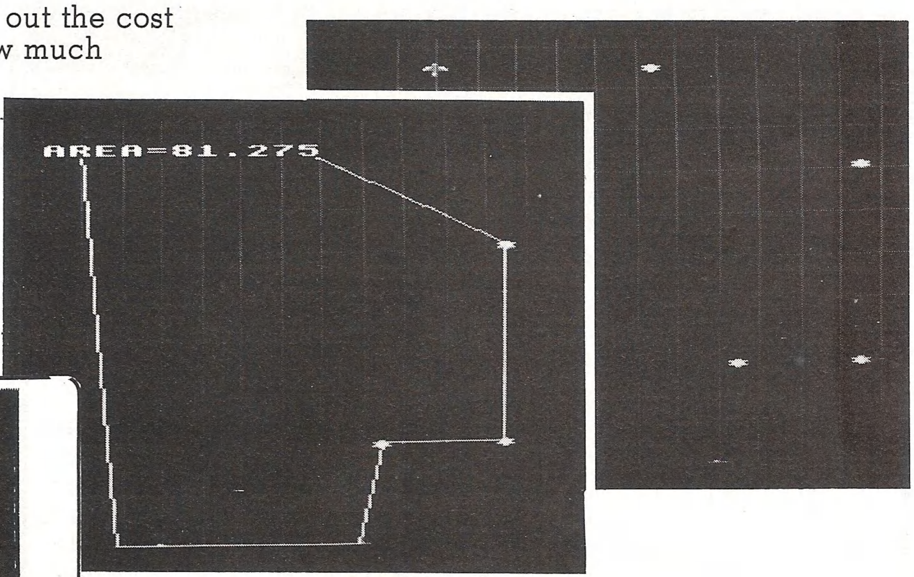
Insert card no. ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Household



Plot 'n' Planner — Work out the cost of your new carpet or how much grass seed you'll need for the lawn. 'Draw' any irregular shape on a displayed grid pattern and the computer will work out the enclosed area.

1.5K



```
AREA=81.275
```

Calendar — With this versatile program you need never be without a date! The program will give the number of days between any two dates, what day of the week any date fell on and a display of any month of any year since 1582!

3K

Mike James

PLOT 'N' PLANNER

Work out the cost of your new carpet or how much grass seed you'll need for the lawn with the help of this ingenious area plotter.

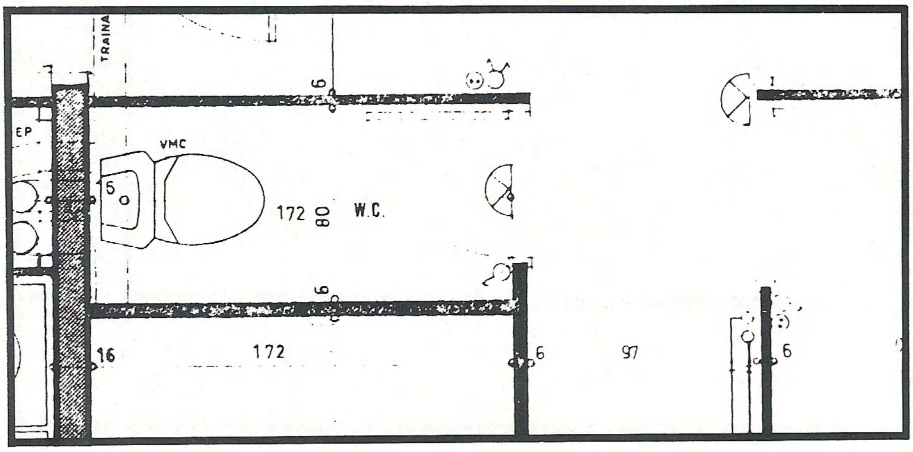
It can take hours to work out the area of an irregular shape but if you are trying to decide which carpet you can afford, how much lino is required to re-do the bathroom or how many paving slabs you'll need for the patio then it's a calculation you need to get right or it could end in disaster! This program does the sums for you and won't mind how many alcoves and window bays you've got to take into account.

A grid appears on the screen and a cross appears in the bottom left. Using the arrow keys you can move this cross around the screen in order to mark points anywhere on it. When you want to mark a point, press the space bar and an asterisk will appear. When you've marked all the points required, the computer joins them up to display the shape of the area you are interested in and calculates the displayed area.

Measuring Up

It's up to you to decide what units you want to measure in — feet, yards or metres. Your chosen unit will, of course, depend on whether you are interested in your own kitchen or the Albert Hall, your back garden or a prairie wheat field. Whatever your chosen measure, let one unit be represented by one interval on the grid.

You may discover lots of other uses for this program. It can help, for example, as you agonise over whether to start tiling the floor from the left-hand or the right-hand corner of the room — draw out both options and look at the consequences, after all it's so easy to rub it out and start again!



PROGRAM STRUCTURE

Statement	Function	Action
Lines 10-70	Set Up	Selects MODE 5, defines grid array, defines character (224) as a cross and enables the arrow keys to be used. The command VDU 5 in line 70 joins the graphics and text areas of the screen.
Lines 80-230	Program Core	Calls the different procedures defined later in the program — that at line 90 prints the grid, that at line 110 plots the cross, and so on. Line 120 registers any keypress, line 140 responds to the space bar being pressed, line 150 responds to 'E' and lines 170 to 200 respond to the arrow keys.
Lines 240-300	Plot	Plots the shape indicated by the vertices.
Lines 310-320	Area	Makes text and graphics independent, calls routine to calculate and print answer.
Lines 330-370	End	Restores normal mode of VDU operation.
Lines 380-450	Calculation	Calculates area and prints result.
Lines 460-530	Plot/unplot	First plots (460-490), then unplots (500-530) the user-controlled cross.
Lines 540-600	Mark	Marks vertices by an asterisk. Emits a sound if unable to accept any vertex.
Lines 610-720	DEFPROC-GRID	Plots grid of lines (in red) on screen. Yellow (colour used to mark vertices) is selected in line 720.

TECHNICAL DETAILS

This program has been written from scratch for the BBC Micro and uses many of its facilities. MODE 5 is selected to allow four colours to be used and the graphics and text screen are joined to allow the cross to be printed anywhere on the screen. The cross character CHR$(224) is defined using a VDU 23 call and is plotted by the usual MOVE X,Y:PRINT CHR$(224); combination. To avoid wiping out either the grid or any plotted asterisks, the cross is plotted after a GCOL 4,0 command which causes the cross to take on the logical NOT of the colour already present at the location. As the cross is unplotted after the same GCOL command, any colours the cross has changed are restored to their original values.

The only other special technique used by this program is the disabling of the cursor editing keys by *FX 4,1. After this command, pressing any cursor key returns an ASCII code instead of moving the cursor, ie they behave like normal keys. To restore the keys to their original function use *FX 4,0.

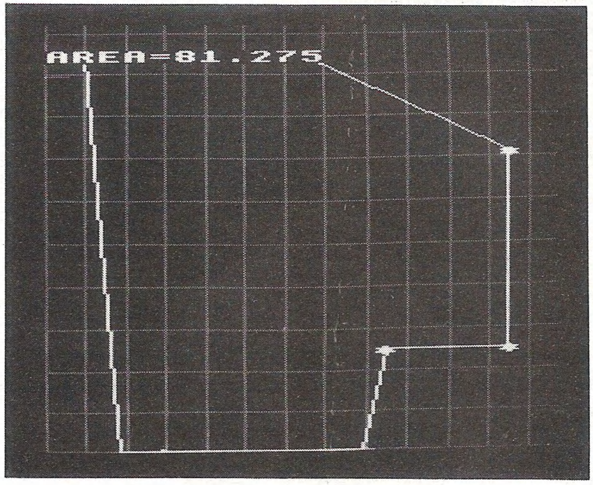
HOW TO USE THE PROGRAM

You can move the pointer (the cross that appears in the bottom left-hand corner) in any direction using the arrow keys. When you want to place a point (vertex) on the grid, press the space bar and a yellow asterisk should then appear. Move the cross to the position you want for your next vertex, mark it with another asterisk and continue until all the vertices are in position. Then type 'E' (and Return) and the vertices will be joined up *in the order in which they were input* and the area will be calculated and displayed.

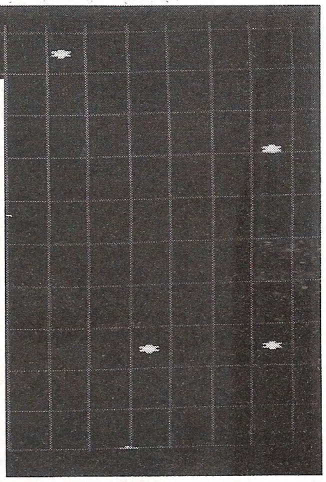
SCOPE FOR IMPROVEMENT

To make the program even more widely applicable, you could allow the user to input a scaling factor — so that one grid interval could represent any number of units you cared to select (eg 1 interval = 5 units, so that 1 square = 25 square units).

Below: The completed plan with its points joined up and the area displayed.



```
AREA=81.275
```



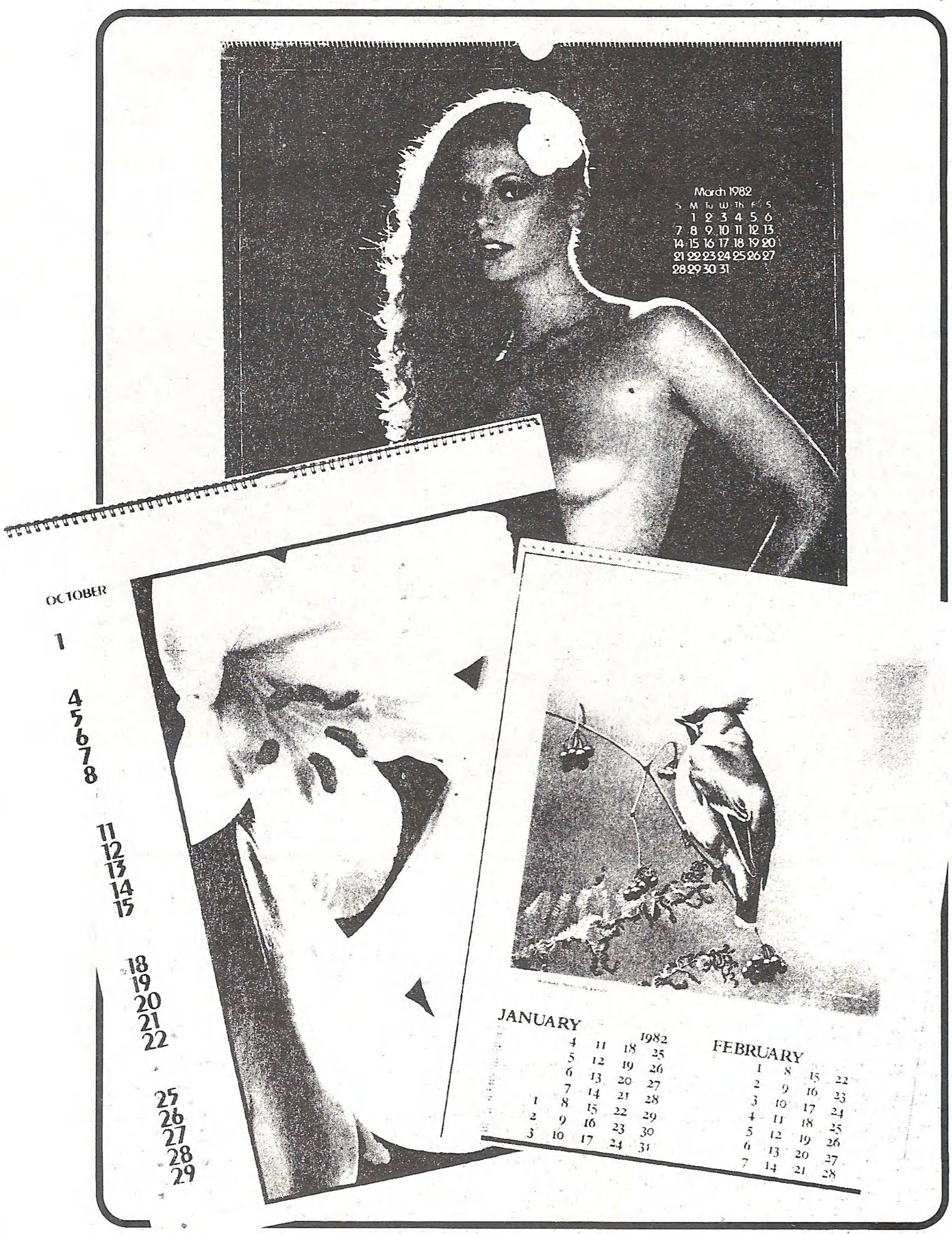
Above: The grid being plotted with the required points.

```
 10  DIM X(50),Y(50)
 20  V=0:Z=0
 30  MODE 5
 40  *FX 4,1
 50  VDU 23,224,&18,&FF,&FF,&18,&18,&18,
     &18,&18
 60  VDU 5
 70  D=10
 80  PROCGRID
 90  X=0:Y=0
100  PROCX(X,Y)
110  A=INKEY(0)
120  OX=X:OY=Y
130  IF A=32 THEN PROCV(X,Y)
140  IF A=69 THEN GOTO 230
150  A=A-&88
160  IF A=0 AND X>0+D THEN X=X-D
170  IF A=1 AND X<1279-D THEN X=X+D
180  IF A=2 AND Y>0+D THEN Y=Y-D
190  IF A=3 AND Y<1023-D THEN Y=Y+D
200  PROCB(OX,OY)
210  PROCX(X,Y)
220  GOTO 110
230  PROCB(X,Y)
240  IF V<3 THEN GOTO 330
250  GCOL 0,3
260  MOVE X(0)+28,Y(0)-8
270  FOR I=1 TO V-1
280  PLOT 5,X(I)+28,Y(I)-8
290  NEXT I
300  PLOT 5,X(0)+28,Y(0)-8
310  VDU 4
320  GOSUB 380
330  A$=GET$
340  MODE 7
350  *FX 4,0
360  VDU 4
370  END
380  Z=0
390  FOR R=0 TO V-1
400  S=R+1-INT((R+1)/V)*V
410  T=R+2-INT((R+2)/V)*V
420  Z=Z+X(S)*(Y(T)-Y(R))
430  NEXT R
440  PRINT TAB(0,2);"AREA=";ABS(Z)/20000
450  RETURN
460  DEFPROCX(X,Y)
470  GCOL 4,0
480  MOVE X,Y:PRINT CHR$(224);
490  ENDPROC
500  DEFPROCB(X,Y)
510  GCOL 4,0
520  MOVE OX,OY:PRINT CHR$(224);
530  ENDPROC
540  DEFPROCV(X,Y)
550  IF V>49THEN SOUND 1,-15,150,10:GOTO 600
560  X(V)=X
570  Y(V)=Y
580  V=V+1
590  MOVE X,Y:PRINT "*"
600  ENDPROC
610  DEFPROCGRID
620  GCOL 0,1
630  FOR X=0 TO 1279 STEP 100
640  MOVE X,0
650  PLOT 5,X,1023
660  NEXT X
670  FOR Y=0 TO 1023 STEP 100
680  MOVE 0,Y
690  PLOT 5,1279,Y
700  NEXT Y
710  GCOL 0,3
720  ENDPROC
```

Mike James

CALENDAR

You need never be at a loss for a date!



This program lets you check your dates in three different ways. It will give you the number of days between any two dates — be they in the present or the past, or in the same or different year of the same or different century. You can use this facility to discover, for example, how many days there are to Christmas, how many days you've already lived or how many days since you left (or started at) school.

It will also tell you the day of the week that any date — in the past five centuries or in the future — will fall on. So you can discover which day of the week you were actually born on or what day your birthday will fall on in ten year's time.

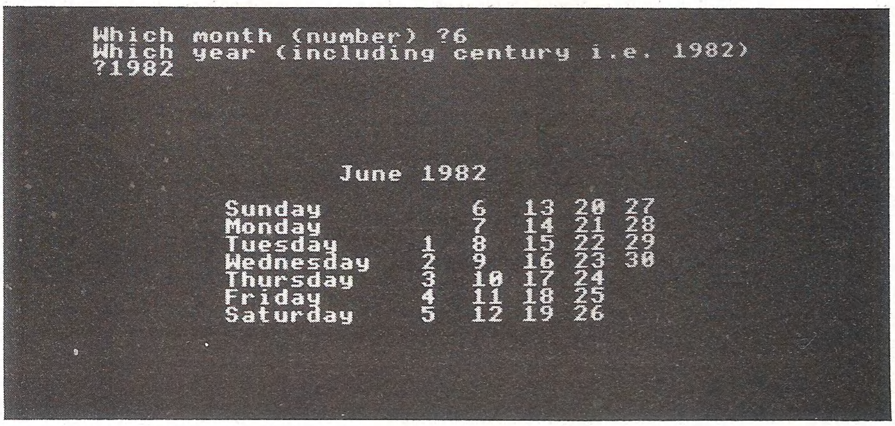
Lastly, it will print out the calendar of any month of any year after 1582 (when the Gregorian Calendar on which we operate came into use). The program takes account of Leap Years — and is not even foxed by the quirk that occurs at the turn of the century.

TECHNICAL DETAILS

This program uses Zeller's Congruence — an equation which will tell you the day of the week for any date taking account of the fact that Leap Years (those years with 29 days in February) occur in every year divisible by four but only in one in four turn of the century years, namely ones where the first two digits are divisible by four.

HOW TO USE THE PROGRAM

This program is self-explanatory. You are asked to choose which option you want — the number of days between any two dates, the day of the week on which a given date falls or to print out the calendar of any month of any year. The important point to remember is to enter the dates which interest



```
Which month (number) ?6
Which year (including century i.e. 1982)
?1982




               June 1982

        Sunday           6  13  20  27
        Monday           7  14  21  28
        Tuesday      1   8  15  22  29
        Wednesday    2   9  16  23  30
        Thursday     3  10  17  24
        Friday       4  11  18  25
        Saturday     5  12  19  26
```

you in number form, specifying first the day (if you have opted for either 1 or 2), then the month and lastly, the year. When inputting the year you are asked to stipulate the century, for example, if you wanted to refer to Christmas Day this year you would write 25,12,1982. If you write 25,12,82 the computer with think you mean the 25th of December in the year 0082!

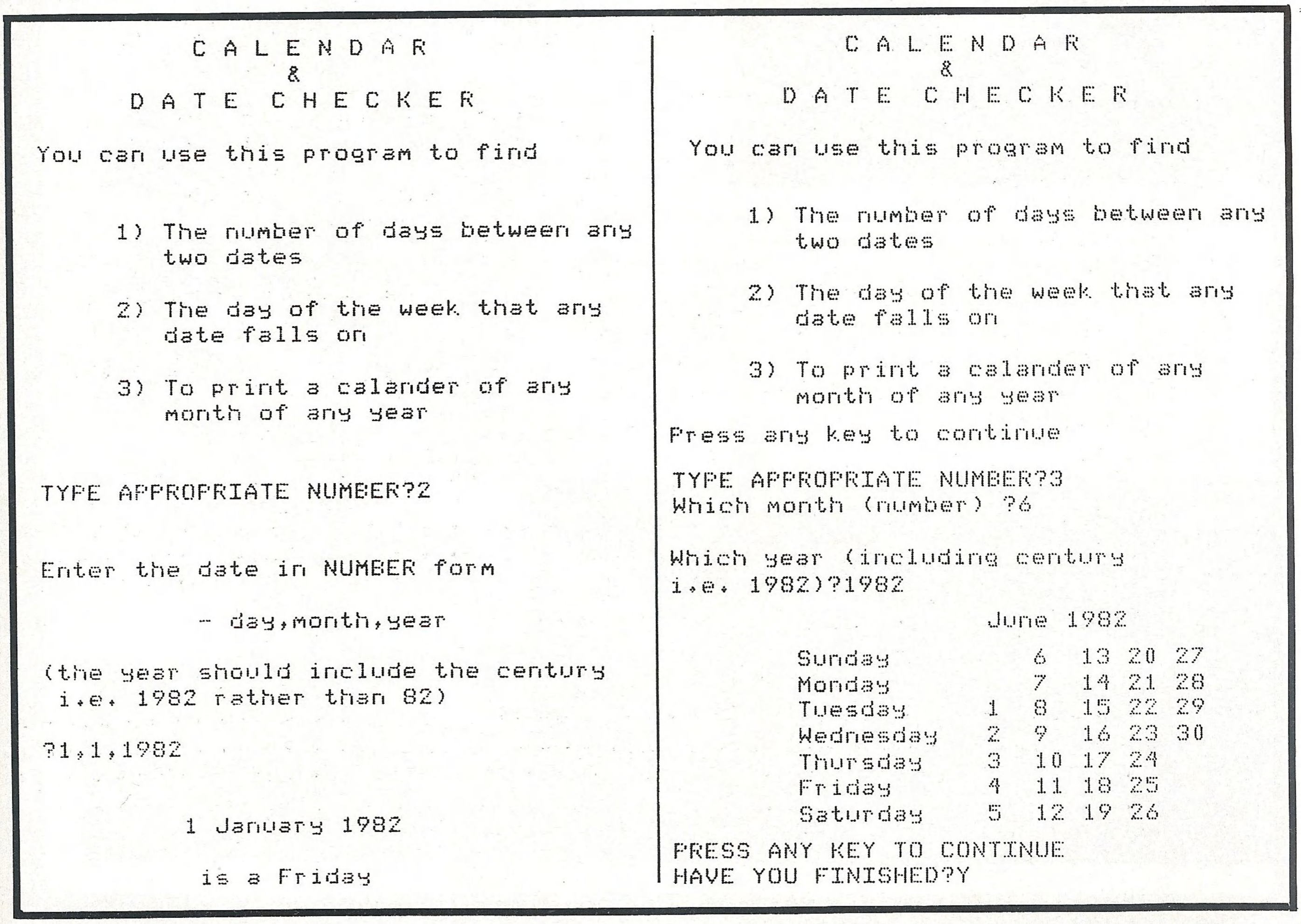
The only limitation on the use of this program is the fact that the Gregorian Calendar, the one we currently use, was only introduced in 1582. If you ask for dates prior to this you will not get an accurate result — but it's still fun to check which day of the week 1,1,1 was!

SCOPE FOR IMPROVEMENT

You could easily modify the program to print the calendar for a full year and by altering the printing format, you could produce a handy month- or year-planner.

PROGRAM STRUCTURE

Statement	Action	Function
Lines 10-30	Set Up	Selects MODE 4. Clears screen and jumps to initialisation routine.
Lines 40-240	Menu	Prints title and instructions with choice of three options.
Lines 250-310	Restart	Checks whether user is finished with program and restarts if required.
Lines 320-440	Days Between	Gets input, calculates the number of days between two dates and prints result.
Lines 450-550	Day of Week	Gets input, calculates which weekday corresponds and prints result.
Lines 560-810	Calendar	Gets input, checks for Leap Year and calculates day of first of month. Lines 710-790 print out calendar.
Lines 820-870	Zeller's Congruence	Calculation to obtain day of the week.
Lines 880-1070	General Date Input	Asks for input of date in prescribed form, checks for validity and returns message if mistake over Leap Year.
Lines 1080-1200	Arrays	Reads in names of months with days in each and names of days.
Lines 1210-1250	Number of Days	Calculates the number of days to a given date.



```
          C A L E N D A R
                &
      D A T E   C H E C K E R

 You can use this program to find


     1) The number of days between any
        two dates

     2) The day of the week that any
        date falls on

     3) To print a calander of any
        month of any year


 TYPE APPROPRIATE NUMBER?2


 Enter the date in NUMBER form

           - day,month,year

 (the year should include the century
   i.e. 1982 rather than 82)

 ?1,1,1982


           1 January 1982

             is a Friday
```

```
          C A L E N D A R
                &
      D A T E   C H E C K E R

 You can use this program to find

     1) The number of days between any
        two dates

     2) The day of the week that any
        date falls on

     3) To print a calander of any
        month of any year
 Press any key to continue

 TYPE APPROPRIATE NUMBER?3
 Which month (number) ?6

 Which year (including century
 i.e. 1982)?1982
                 June 1982

      Sunday         6  13 20 27
      Monday         7  14 21 28
      Tuesday     1  8  15 22 29
      Wednesday   2  9  16 23 30
      Thursday    3  10 17 24
      Friday      4  11 18 25
      Saturday    5  12 19 26
 PRESS ANY KEY TO CONTINUE
 HAVE YOU FINISHED?Y
```

```
10   MODE 4
20   CLS
30   PROCINIT
40   PRINT:PRINT
50   PRINT TAB(10);"C A L E N D A R"
60   PRINT TAB(16);"&"
70   PRINT TAB(6);"D A T E  C H E C K E R"
80   PRINT TAB(0,7)
90   PRINT"You can use this program to find"
100  PRINT TAB(0,9)
110  PRINT
120  PRINT TAB(5);"1) The number of
     days between any"
130  PRINT TAB(5);"   two dates"
140  PRINT
150  PRINT TAB(5);"2) The day of the
     week that any"
160  PRINT TAB(5);"   date falls on"
170  PRINT
180  PRINT TAB(5);"3) To print a
     calander of any"
190  PRINT TAB(5);"   month of any year"
200  PRINT:PRINT
210  INPUT"TYPE APPROPRIATE NUMBER",A
220  IF A<1 OR A>3 THEN PRINT TAB(0,25);
     "TYPE A NUMBER BETWEEN ONE AND THREE";
     TAB(0,20):GOTO 210
240  ON A GOSUB 320,450,560
250  CLS
260  PRINT:INPUT "HAVE YOU FINISHED",A$
270  A$=LEFT$(A$,1)
280  IF A$="N" THEN RUN
290  IF A$<>"Y" THEN GOTO 250
300  CLS
310  END
320  CLS:PRINT:PRINT
330  PROCDATE
340  FOR J=1 TO 12
350  M(J)=M(J)+M(J-1)
360  NEXT J
370  PROCDAYS
380  D1=DAY
390  PROCDATE
400  PROCDAYS
410  PRINT:PRINT "The number of days
     between the"
420  PRINT"two dates is ";ABS(D1-DAY)
430  PRINT TAB(0,31);"PRESS ANY KEY TO
     CONTINUE";:A$=GET$
440  RETURN
450  CLS
460  PRINT:PROCDATE
470  PROCZEL(DAY,MONTH,YEAR,CEN)
480  PRINT
490  PRINT
500  PRINT DAY;" ";M$(MONTH);" ";CEN;YEAR
510  PRINT
520  PRINT TAB(10); "is a ";DAY$(I)
530  PRINT
540  PRINT"Press any key to continue":A$=GET$
550  RETURN
560  CLS
570  PRINT:PRINT
580  INPUT "Which month (number) ",MONTH
590  IF MONTH<1 OR MONTH>12 THEN GOTO 580
600  INPUT "Which year (including
     century i.e. 1982)",YEAR
610  FOR I=1 TO 1000:NEXT I
620  DAY=1
630  CEN=INT(YEAR/100)
640  YEAR=YEAR-CEN*100
650  LEAP=0
660  IF (YEAR-INT(YEAR/4)*4)=0 OR
     (CEN-INT(CEN/4)*4)=0 THEN LEAP=1
670  PRINT TAB(15,10);M$(MONTH);" ";CEN;YEAR
680  PROCZEL(DAY,MONTH,YEAR,CEN)
690  IF LEAP=1 THEN M(2)=29
700  PRINT
710  FOR J=0 TO 6
720  PRINT TAB(8);DAY$(J);
730  FOR K=0 TO 5
740  D=K*7+J+1-I
750  IF D<1 OR D>M(MONTH) THEN GOTO 770
760  PRINT TAB(K*3+20);D;
770  NEXT K
780  PRINT
790  NEXT J
800  PRINT TAB(0,30);"PRESS ANY KEY TO
     CONTINUE";:A$=GET$
810  RETURN
820  DEFPROCZEL(DAY,MONTH,YEAR,CEN)
830  MONTH=MONTH-2
840  IF MONTH<1 THEN MONTH=MONTH+12:YEAR=YEAR-1
850  I=INT(2.6*MONTH-.19)+DAY+YEAR+
     INT(YEAR/4)+INT(CEN/4)-2*CEN
860  I=I-INT(I/7)*7
870  ENDPROC
880  DEFPROCDATE
890  PRINT
900  PRINT "Enter the date in NUMBER form"
910  PRINT
920  PRINT TAB(10);"- day,month,year"
930  PRINT
940  PRINT"(the year should include the
     century"
950  PRINT" i.e. 1982 rather than 82)"
960  PRINT
970  INPUT DAY,MONTH,YEAR
980  LEAP=0
990  CEN=INT(YEAR/100)
1000 YEAR=YEAR-CEN*100
1010 IF (YEAR-INT(YEAR/4)*4)=0 OR
     (CEN-INT(CEN/4)*4=0) THEN LEAP=1
1020 IF M(MONTH)>DAY AND DAY>0 THEN GOTO 1070
1030 IF MONTH=2 AND LEAP=1 AND DAY=29
     THEN GOTO 1070
1040 PRINT
1050 PRINT "There is no such day in ";
     M$(MONTH)
1060 GOTO 900
1070 ENDPROC
1080 DEF PROCINIT
1090 DIM M$(12),M(12),DAY$(6)
1100 DATA January,31,February,28,March,31,
     April,30,May,31
1110 DATA June,30,July,31,August,31,
     September,30
1120 DATA October,31,November,30,December,31
1130 DATA Sunday,Monday,Tuesday,
     Wednesday,Thursday,Friday,Saturday
1140 FOR I=1 TO 12
1150 READ M$(I),M(I)
1160 NEXT I
1170 FOR I=0 TO 6
1180 READ DAY$(I)
1190 NEXT I
1200 ENDPROC
1210 DEFPROCDAYS
1220 YEAR=YEAR+CEN*100
1230 DAY=DAY+M(MONTH)+365*YEAR+INT((YEAR-1)/4)
1240 IF INT(YEAR/4)*4=YEAR AND MONTH>2
     THEN DAY=DAY-1
1250 ENDPROC
```

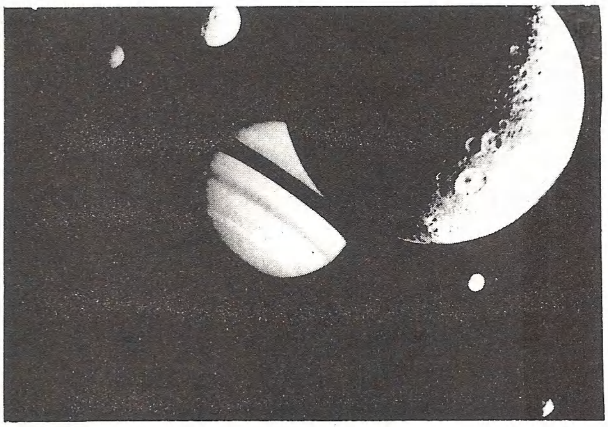
Educational

Descriptive Statistics — As statistics packages go, this is a short and simple one — but it will still turn unwieldy raw data into neat statistical information. **5.5K**

OES-4
80:268:18:12 A1

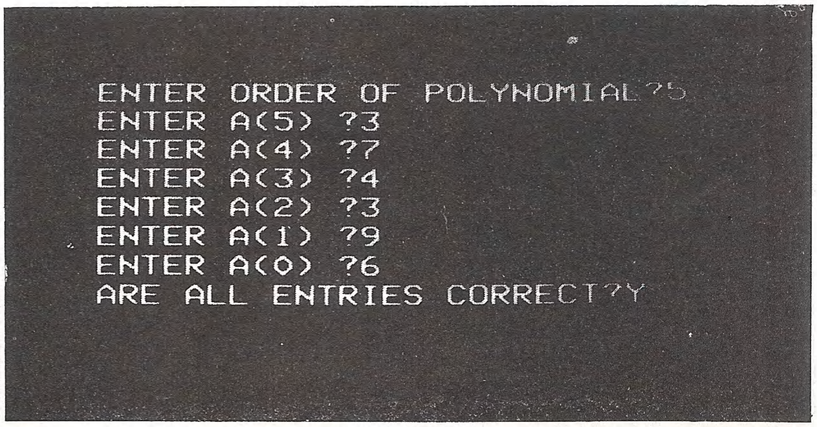
Newton's Law — With this program, the computer enables the user to get the feel for Newton's Law of Gravitation as applied to the motion of planets. The orbits of planets are calculated and displayed on the screen. It can even simulate the 'slingshot' effect as used to send the Voyager spacecraft from Jupiter to Saturn!

4.5K



Root Solver — This program will calculate and list the roots of any polynomial (providing the coefficients are real!). The polynomial can be of any order and both real and complex roots are derived.

2.5K



```
ENTER ORDER OF POLYNOMIAL?5
ENTER A(5) ?3
ENTER A(4) ?7
ENTER A(3) ?4
ENTER A(2) ?3
ENTER A(1) ?9
ENTER A(0) ?6
ARE ALL ENTRIES CORRECT?Y
```

Converted from a program by A P Stephenson

DESCRIPTIVE STATISTICS

Get to grips with statistics on your BBC Micro.

PROGRAM STRUCTURE

Statement	Function	Action
Lines 10-30	Set Up	Selects MODE 4, defines character 224 as a solid square and defines array D.
Lines 40-240	Menu	Displays the nine options available.
Lines 250-280	Selection	User asked to choose option, jumps to appropriate section of program.
Lines 290-510	Input Data	User allowed to input data or to add data to existing file.
Lines 520-600	Name File	User asked to name data file.
Lines 610-730	Random Data	Computer generates random data file to user's specifications.
Lines 740-860	Data Display	Data file listed, seventeen items at a time.
Lines 870-970	Edit	User allowed to modify any data item.
Lines 980-990	Add Items	Jumps to line 390 if option 5 requested.
Lines 1000-1190	Bar Chart	Displays data items as a bar chart (first 38 items only).
Lines 1200-1440	Calculations	Calculates sum and mean of data items, variance and standard deviation. Notes maximum value.
Lines 1450-1520	Results	Prints results of the above.
Lines 1530-1670	Confidence Limits	Indicates confidence intervals for 'normal' data.
Lines 1680-1700	Continue	'Press any key to continue' message.
Lines 1710-1730	Input	General input routine.
Lines 1740-1800	Print Histogram	Produces histogram of first 38 data items.
Lines 1810-1830	Y-axis	Prints y-axis for bar chart.
Lines 1840-1880	Maximum	Finds maximum value among data items.
Lines 1890-1980	Borders	Prints borders to tabular displays.
Lines 1990-2110	Save File	Instructions to save data file on tape with routine.
Lines 2120-2280	Load File	Instructions to load data file from tape with routine.

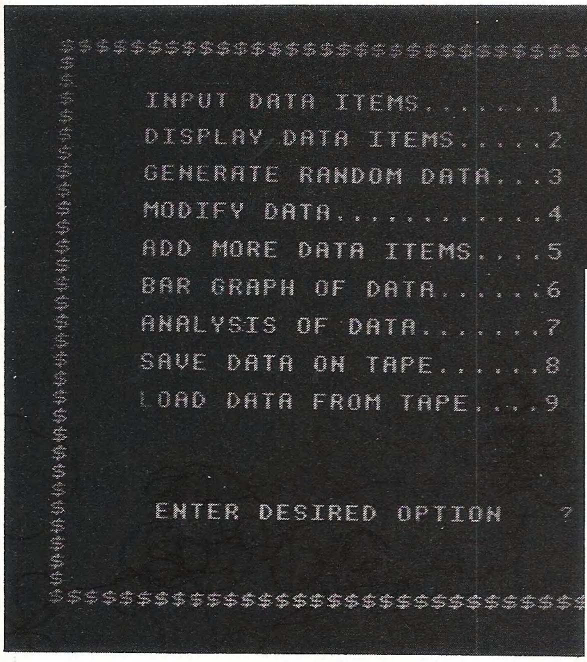
As statistics packages go, this is a short and simple one. But, it will still turn unwieldy raw data into neat statistical information.

Nowadays everybody has an opinion about statistics — you either hate and fear them or you revel in them. To the cynic, a statistic is a damned lie, to the politician a statistic is a white lie and to the scientist and engineer, a statistic is a mathematically acceptable guess used as a last resort when knowledge is either feeble or too expensive to obtain!

Unfortunately, statistics is a branch of mathematics which even its devotees can find tiresome. It promises so much but even simple exercises involve the manipulation of a depressing amount of data. The beginner is often deterred from trying to master the subject simply because it takes so long to get an answer. This state of affairs obviously calls for a computer program.

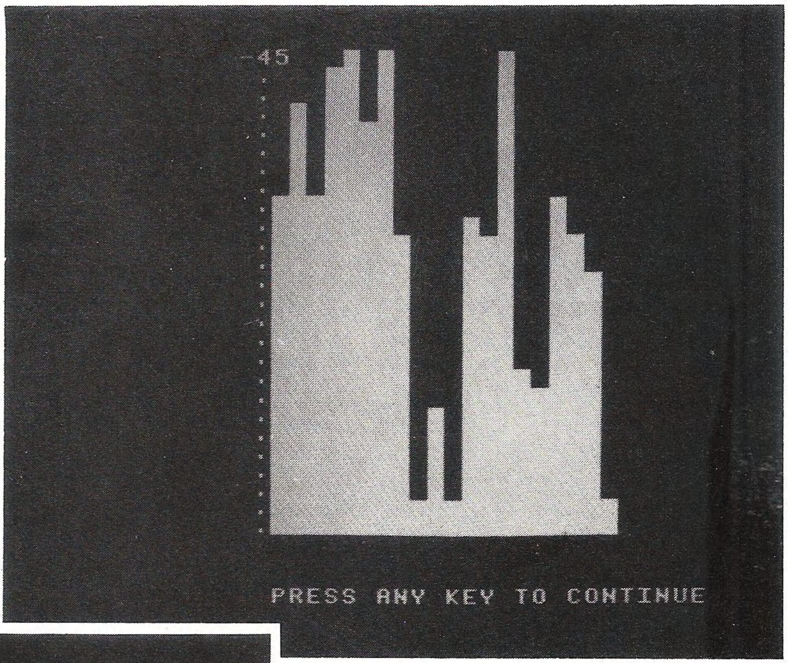
Statistically Speaking

This particular program provides an easy method of entertaining and modifying statistical data and would certainly help anyone to try out the various lengthy examples which appear to be vital in gaining confidence in the field of statistics. The actual analysis of the data is the comparatively easy part and can be expanded according to the knowledge and requirements of individuals. This program could therefore be considered a data input skeleton which could form a useful starting point for a more extensive one.



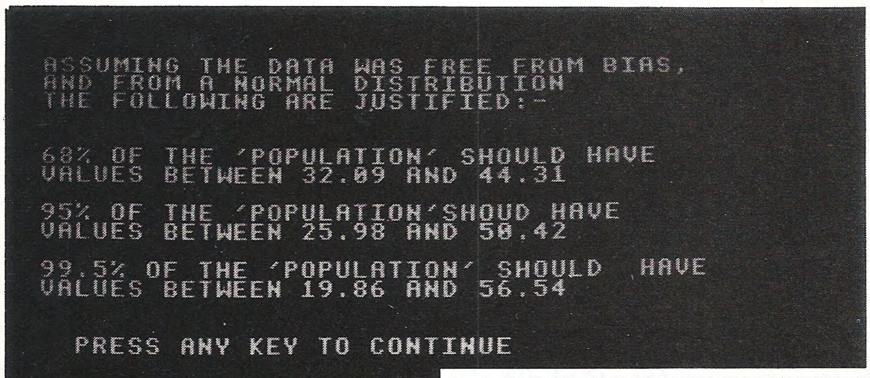
```
INPUT DATA ITEMS.......1
DISPLAY DATA ITEMS.....2
GENERATE RANDOM DATA...3
MODIFY DATA............4
ADD MORE DATA ITEMS....5
BAR GRAPH OF DATA......6
ANALYSIS OF DATA.......7
SAVE DATA ON TAPE......8
LOAD DATA FROM TAPE....9


ENTER DESIRED OPTION    ?
```

Left: The main menu generated by the program. The information analysed below has been generated by Option 3, the random data generator.



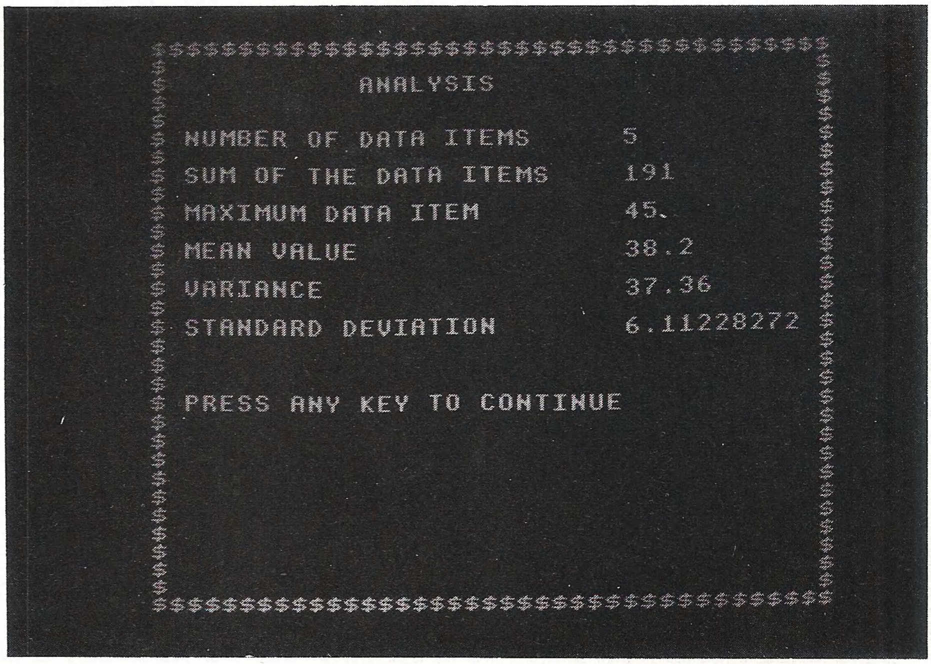
```
PRESS ANY KEY TO CONTINUE
```

Right: A bar-chart display of the random data. Up to 38 entries can be displayed across the screen.



```
ASSUMING THE DATA WAS FREE FROM BIAS,
AND FROM A NORMAL DISTRIBUTION
THE FOLLOWING ARE JUSTIFIED:-

68% OF THE 'POPULATION' SHOULD HAVE
VALUES BETWEEN 32.09 AND 44.31

95% OF THE 'POPULATION'SHOUD HAVE
VALUES BETWEEN 25.98 AND 50.42

99.5% OF THE 'POPULATION' SHOULD  HAVE
VALUES BETWEEN 19.86 AND 56.54

   PRESS ANY KEY TO CONTINUE
```

Left: The analysis of the data is performed between specified confidence limits.



```
              ANALYSIS

NUMBER OF DATA ITEMS        5
SUM OF THE DATA ITEMS       191
MAXIMUM DATA ITEM           45
MEAN VALUE                  38.2
VARIANCE                    37.36
STANDARD DEVIATION          6.11228272

PRESS ANY KEY TO CONTINUE
```

Right: The final analysis of the data provides the usual statistical information.

TECHNICAL DETAILS

This program has been converted from PET BASIC. The only interesting features of the conversion are the use of CHR$(224) to produce the histogram bars and the use of TAB(X,VPOS) to tabulate across the current line without wiping out the borders (lines 70-250).

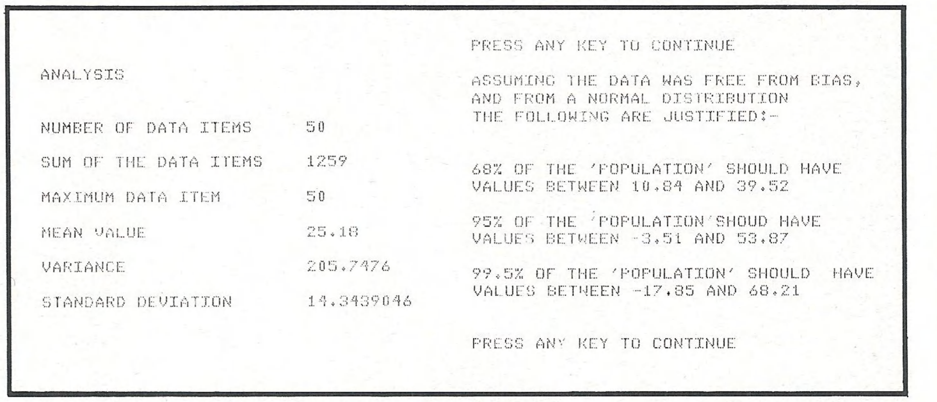
HOW TO USE THE PROGRAM

The program gives you nine options but you will have to choose either option 1 (input data) or option 3 (generate random data) on the first occasion you use it. Having obtained some data you can then choose to display it (option 2), modify it (option 4) or add to it (option 5).

Option 6 enables you to display a bar graph of your data. This is only likely to be useful with very special sorts of data as the number of items it can handle is limited to 38 and each data item is displayed separately. This is NOT a histogram for displaying frequencies although it would be useful with time series data — for example, showing monthly rainfall or for hopeful slimmers, the number of pounds lost per week.

For a statistical analysis of your data, select option 7. The computer will then give you the sum of all your data items and their mean, the maximum value of the items and their variance and standard deviation. It will also display 'confidence limits' for the data.

The program has one other useful facility — that of saving your data onto a cassette (option 8) and re-loading it (option 9).



```
                                           PRESS ANY KEY TO CONTINUE

ANALYSIS                                   ASSUMING THE DATA WAS FREE FROM BIAS,
                                           AND FROM A NORMAL DISTRIBUTION
NUMBER OF DATA ITEMS      50               THE FOLLOWING ARE JUSTIFIED:-

SUM OF THE DATA ITEMS     1259             68% OF THE ‘POPULATION’ SHOULD HAVE
                                           VALUES BETWEEN 10.84 AND 39.52
MAXIMUM DATA ITEM         50
                                           95% OF THE ‘POPULATION’SHOUD HAVE
MEAN VALUE                25.18            VALUES BETWEEN -3.51 AND 53.87

VARIANCE                  205.7476         99.5% OF THE ‘POPULATION’ SHOULD  HAVE
                                           VALUES BETWEEN -17.85 AND 68.21
STANDARD DEVIATION        14.3439046

                                           PRESS ANY KEY TO CONTINUE
```

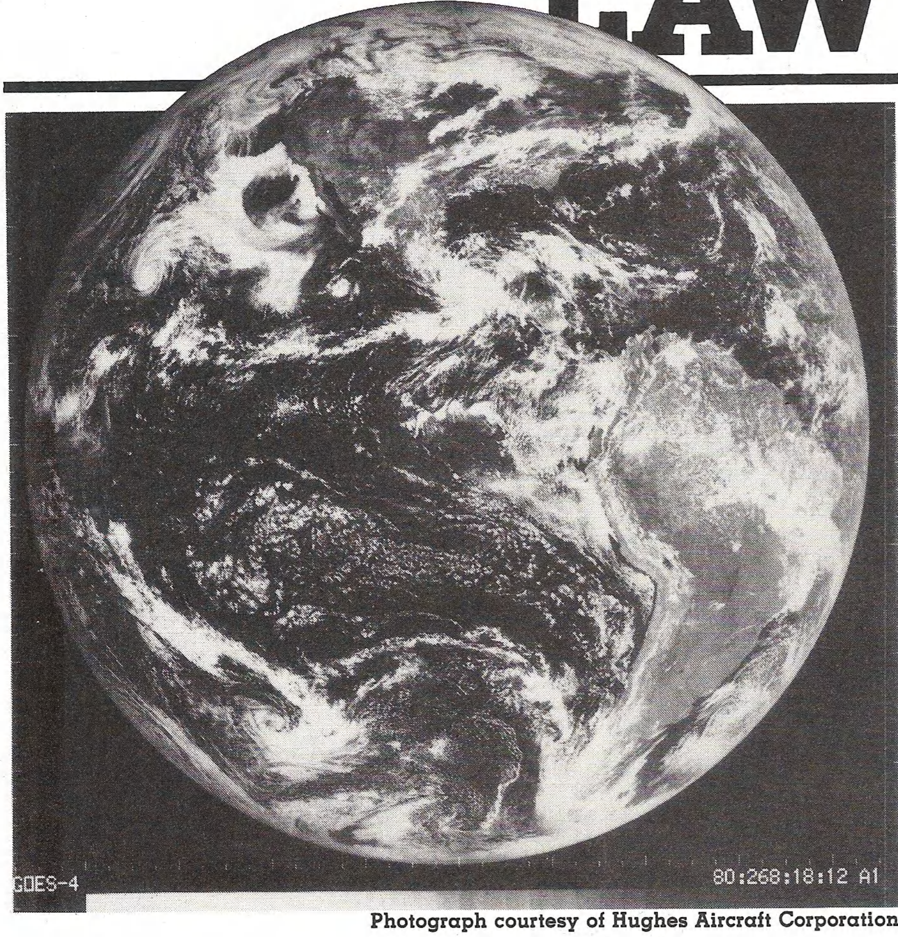
```
 10  MODE 4
 20  VDU 23,224,&FF,&FF,&FF,&FF,&FF,&FF,
     &FF,&FF
 30  DIM D(1000)
 40  GOSUB 1910:REM**CLEARS SCREEN
 50  VDU 30
 60  PRINT:PRINT:PRINT
 70  PRINT TAB(5,VPOS);"INPUT DATA
     ITEMS.......1"
 80  PRINT
 90  PRINT TAB(5,VPOS);"DISPLAY DATA
     ITEMS.....2"
100  PRINT
110  PRINT TAB(5,VPOS);"GENERATE
     RANDOM DATA...3"
120  PRINT
130  PRINT TAB(5,VPOS);"MODIFY
     DATA............4"
140  PRINT
150  PRINT TAB(5,VPOS);"ADD MORE DATA
     ITEMS....5"
160  PRINT
170  PRINT TAB(5,VPOS);"BAR GRAPH OF
     DATA......6"
180  PRINT
190  PRINT TAB(5,VPOS);"ANALYSIS OF
     DATA.......7"
200  PRINT
210  PRINT TAB(5,VPOS);"SAVE DATA ON
     TAPE......8"
220  PRINT
230  PRINT TAB(5,VPOS);"LOAD DATA FROM
     TAPE....9"
240  PRINT
250  PRINT TAB(6,25);"ENTER DESIRED
     OPTION  ";
260  GOSUB 1720:K$=I8$
270  IF VAL(K$)<1 OR VAL(K$)>11 THEN 260
280  ON VAL(K$) GOTO 290,740,610,870,
     980,1000,1210,1990,2120
290  CLS:PRINT
300  PRINT SPC(7);"NOTE THE FOLLOWING :
     -":PRINT
310  PRINT
320  PRINT TAB(3);"1. DATA MUST BE
     NUMERIC":PRINT:PRINT
330  PRINT TAB(3);"2. PRESS RETURN
     AFTER EACH ITEM":PRINT:PRINT
340  PRINT TAB(3);"3. KEY END TO
     TERMINATE INPUT":PRINT
350  PRINT
360  GOSUB 1680
370  CLEAR:DIM D(1000):E$="ENTER ":
     IT$="ITEM NUMBER "
380  CLS:F=0:N=1:REM**FLAG FOR ADD DATA OPTION
390  IF F=1 THEN N=NM+1
400  PRINT
410  E$="ENTER ":IT$="ITEM NUMBER "
420  PRINT E$+IT$;N; TAB(26);
430  GOSUB 1710
440  PRINT
450  IF I8$="END" THEN N=N-1:NM=N:GOTO 510
460  D(N)=VAL(I8$):N=N+1
470  IF N-1=1 THEN DM=D(N-1)
480  IF D(N-1)>DM THEN DM=D(N-1)
490  F=0
500  GOTO 410
510  PRINT
520  CLS:PRINT:PRINT TAB(240)
530  PRINT "THE DATA FILE MUST BE
     GIVEN A NAME":PRINT
540  PRINT "THE NUMBER OF CHARACTERS
     MUST BE":PRINT
550  PRINT TAB(12);"LESS THAN ELEVEN"
560  PRINT TAB(12,15);
570  PRINT "ENTER THE NAME NOW";
580  GOSUB 1720
590  TN$=I8$
600  GOTO 40
610  CLS:PRINT TAB(240):DM=1:X=RND(1)
620  PRINT "HOW MANY RANDOM INTEGERS ";
630  PRINT TAB(30);:GOSUB 1720:
     NM=VAL(I8$):PRINT:PRINT
640  PRINT "LOWEST INTEGER WANTED ";
650  PRINT TAB(30);:GOSUB 1720:
     M=VAL(I8$):PRINT:PRINT
660  PRINT "HIGHEST INTEGER WANTED ";
670  PRINT TAB(30);:GOSUB 1720:
     H=VAL(I8$):PRINT:PRINT:PRINT:PRINT
680  PRINT TAB(10);"WAIT A FEW SECONDS "
690  FOR N=1 TO NM
700  D(N)=INT((H-M+1)*RND(1)+M)
710  IF D(N-1)>DM THEN DM=D(N-1)
720  NEXT N
730  GOTO 520
740  CLS:IT$="ITEM NUMBER "
```

```
750  PRINT TAB(10);"DATA HELD ON FILE":
     PRINT:PRINT
760  PRINT TAB(6);"DATA IS NAMED ";TN$:PRINT
770  P=0
780  FOR S=1 TO NM
790  IF P=17 THEN PRINT:GOSUB 1680:P=1:
     CLS:PRINT
800  PRINT IT$;S;TAB(25);D(S)
810  P=P+1
820  NEXT S
830  PRINT
840  PRINT:PRINT TAB(13)"END OF DATA"
850  P=0:F=0
860  PRINT:GOSUB 1680:GOTO 40
870  CLS
880  PRINT TAB(5);"MODIFY SELECTED
     DATA ITEM":PRINT:PRINT:PRINT
890  PRINT "ENTER DATA ITEM NUMBER ";:
     GOSUB 1710:PRINT
900  IF VAL(I8$)<=NM THEN PRINT:GOTO 920
910  PRINT:PRINT TAB(5);"NO SUCH DATA
     ITEM":PRINT:GOTO 890
920  I8=VAL(I8$) :PRINT:PRINT
930  PRINT "DATA ITEM ";I8;" IS ";
     D(I8):PRINT:PRINT:PRINT
940  PRINT "ENTER THE NEW VALUE ";:
     GOSUB  1710
950  D(I8)=VAL(I8$)
960  GOSUB 1840
970  GOTO 40
980  CLS:F=1:GOTO 390
990  PRINT "ENTER THIS NOW...";
1000 CLS:PRINT TAB(240)
1010 PRINT "DUE TO LIMITATIONS IMPOSED
     BY THE SCREEN":PRINT
1020 PRINT "WIDTH, THE BAR GRAPH ONLY
     DISPLAYS THE":PRINT
1030 PRINT "FIRST 38 DATA ITEMS":PRINT:PRINT
1040 PRINT:PRINT:PRINT
1050 GOSUB 1690
1060 CLS:GOSUB 1840
1070 PRINT TAB(0,1);"-";DM
1080 N=1
1090 FOR X8=2 TO 39
1100 IF DM=0 THEN DM=1E-30
1110 N8=25*D(N)/DM
1120 GOSUB 1740
1130 N=N+1
1140 NEXT X8
1150 GOSUB 1810
1160 PRINT TAB(0,30);
1170 GOSUB 1680
1180 N=N-1
1190 GOTO 40
1200 REM**CALCULATIONS ON THE DATA
1210 CLS:V=0:T=0
1220 PRINT:PRINT:PRINT:PRINT:PRINT
1230 PRINT TAB(9)"ANALYSIS PROCEEDING"
1240 IF NM=0 THEN NM=1E-30
1250 FOR N=1 TO NM:T=T+D(N):NEXT
1260 ME=T/NM
1270 FOR N=1 TO NM
1280 V=V+(D(N)-ME)^2
1290 NEXT
1300 V=V/NM
1310 SD=SQR(V)
1320 L1=ME-SD:H1=ME+SD
1330 L1=INT(100*L1+.5)/100
1340 H1=INT(100*H1+.5)/100
1350 L2=ME-2*SD:H2=ME+2*SD
1360 L2=INT(100*L2+.5)/100
1370 H2=INT(100*H2+.5)/100
1380 L3=ME-3*SD:H3=ME+3*SD
1390 L3=INT(100*L3+.5)/100
1400 H3=INT(100*H3+.5)/100
1410 CLS:GOSUB 1910:VDU 30:PRINT:PRINT:
     PRINT TAB(12,VPOS);"ANALYSIS"
1420 PRINT
1430 IF NM=1E-30 THEN NM=0
1440 IF DM=1E-30 THEN DM=0
1450 PRINT
1460 PRINT TAB(2,VPOS);"NUMBER OF DATA
     ITEMS";TAB(25);NM:PRINT
1470 PRINT TAB(2,VPOS);"SUM OF THE
     DATA ITEMS";TAB(25);T:PRINT
1480 PRINT TAB(2,VPOS);"MAXIMUM DATA
     ITEM";TAB(25);DM:PRINT
1490 PRINT TAB(2,VPOS);"MEAN VALUE";
     TAB(25);ME:PRINT
1500 PRINT TAB(2,VPOS);"VARIANCE";
     TAB(25);V:PRINT
1510 PRINT TAB(2,VPOS);"STANDARD
     DEVIATION";TAB(25);SD:PRINT:PRINT:PRINT
1520 GOSUB 1690
1530 CLS:PRINT
1540 PRINT "ASSUMING THE DATA WAS FREE
     FROM BIAS,"
1550 PRINT "AND FROM A NORMAL DISTRIBUTION"
1560 PRINT "THE FOLLOWING ARE JUSTIFIED:-"
1570 PRINT:PRINT
1580 PRINT "684 OF THE 'POPULATION'
     SHOULD HAVE"
1590 PRINT "VALUES BETWEEN ";L1;" AND "H1
1600 PRINT:PRINT "954 OF THE
     'POPULATION'SHOUD HAVE"
1610 PRINT "VALUES BETWEEN ";L2;" AND ";H2
1620 PRINT
1630 PRINT "99.54 OF THE 'POPULATION'
     SHOULD  HAVE"
1640 PRINT "VALUES BETWEEN ";L3;" AND  ";H3
1650 PRINT:PRINT
1660 GOSUB 1690
1670 GOTO 40
1680 REM**ANY KEY ROUTINE
1690 PRINT TAB(2,VPOS);"PRESS ANY KEY
     TO CONTINUE":A$=GET$
1700 RETURN
1710 REM INPUT
1720 INPUT I8$
1730 RETURN
1740 REM**VERTICAL BAR OF N8
1750 REM**AT CO-ORDINATE X8
1760 IF N8>25 THEN N8=25
1770 FOR Z8=26 TO 26-N8 STEP -1
1780 IF N8=0 THEN 1800
1790 PRINT TAB(X8,Z8);CHR$(224);
1800 NEXT:RETURN
1810 FOR C8=2 TO 26
1820 PRINT TAB(1,C8);".";
1830 NEXT:RETURN
1840 DM=D(1):REM**FIND THE MAX VALUE
1850 FOR N=1 TO NM-1
1860 IF D(N+1)>DM THEN DM=D(N+1)
1870 NEXT N
1880 RETURN
1890 GOTO 40
1900 REM **USE $$$ TO
1910 REM**PRINT BORDERS
1920 CLS
1930 PRINT STRING$(39,"$")
1940 FOR L=1 TO 29
1950 PRINT "$";TAB(38,L);"$"
1960 NEXT
1970 PRINT STRING$(39,"$")
1980 RETURN
1990 CLS:PRINT
2000 PRINT TAB(3);"1. INSERT A CASSETTE":PRINT
2010 PRINT TAB(3);"2. REWIND IT":PRINT
2020 PRINT TAB(3);"3. PRESS 'PLAY' AND
     'RECORD'":PRINT:PRINT
2030 IF TN$="" THEN TN$="NONAME"
2040 FILE=OPENOUT(TN$)
2050 PRINT#FILE,NM
2060 PRINT#FILE,DM
2070 FOR N=1 TO NM
2080 PRINT#FILE,D(N)
2090 NEXT N
2100 CLOSE #FILE
2110 GOTO 40
2120 CLS:PRINT
2130 PRINT "WHAT IS THE NAME OF THE
     FILE ";:PRINT
2140 CLEAR:DIM D(1500)
2150 GOSUB 1710:PRINT:PRINT
2160 TN$=I8$
2170 PRINT TAB(3);"1. INSERT THE DATA
     TAPE":PRINT
2180 PRINT TAB(3);"2. REWIND IT":PRINT
2190 PRINT TAB(3);"3. PRESS PLAY"
2200 PRINT:PRINT
2210 FILE=OPENIN(TN$)
2220 INPUT#FILE,NM
2230 INPUT#FILE,DM
2240 FOR N=1 TO NM
2250 INPUT#1,D(N)
2260 NEXT N
2270 CLOSE #FILE
2280 GOTO 40
```

Converted from a program by J D Lee and T D Lee.

NEWTON'S LAW

Observe the effects of gravitational force on your BBC Micro.



Photograph courtesy of Hughes Aircraft Corporation

This program allows the user to get a feel for Newton's Law of Gravitation as applied to the motion of planets. The orbits of planets are calculated and displayed and by experimentation the user can find stable orbits (circles or ellipses), and the precession of elliptical orbits may be observed.

It is also possible to simulate a 'slingshot' such as that used to send the Voyager spacecraft from Jupiter to Saturn. In this the spacecraft is aimed in front of the planet, loops round the planet due to its gravity and finally leaves the planet moving faster than before the encounter. By this devious route the spacecraft gets to Saturn quicker than by going directly — an ingenious way of saving time and fuel!

The Final Frontier

Tycho Brahe was one of the earliest astronomers to make accurate observations of the planets. Using his data, Kepler deduced three laws of planetary motion:

1. Planets move in ellipses, with the sun at one focus.
2. The radius vector joining the planet to the sun sweeps out equal areas in equal time.
3. The period of orbit squared is directly proportional to the semi-major axis cubed.

Robert Hooke, who had been studying the effects of gravity, suggested that gravity might vary with the inverse square of distance. Newton checked Hooke's ideas and concluded that an inverse square law for gravity would result in elliptical planetary orbits.

An Apple A Day

Newton's name is remembered since he was the first to publish the result now generally known as Newton's Law of Gravitation:

$$F = \frac{G \times M1 \times M2}{d^2}$$

where F is the force of attraction between two bodies (planet and sun), M1 and M2 are the masses of the two bodies, d is the distance between the bodies and G is the universal gravitational constant. The value of G depends on the system of units used; with SI units:

$G = (6.673 \pm 0.003 \times 10^{-17}$ Newton metres^2/kilogram^2

The program uses the equation for Newton's Law of Gravitation together with Newton's second Law of Motion which states:

force acting on a body = mass of body x acceleration of body

or

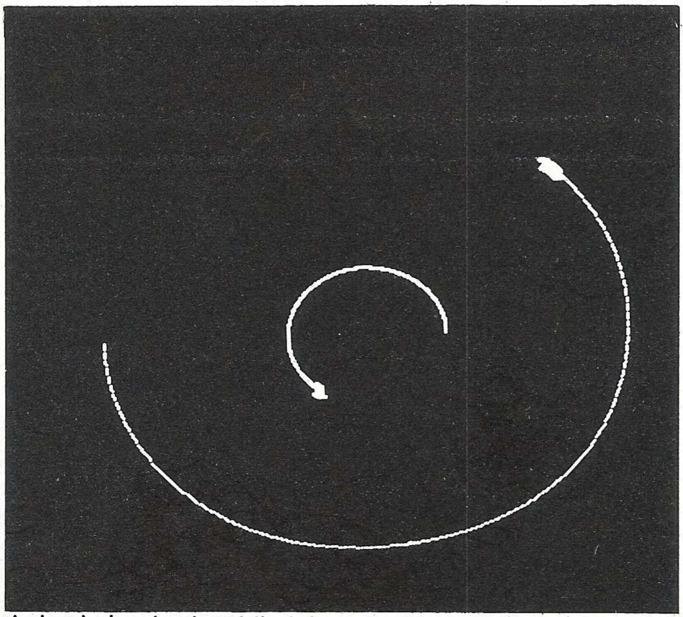
$F = M \times a$

Very accurate measurements of the orbit of the planet Mercury show that it precesses slightly faster than can be explained using Newton's Laws, and this is explained by Einstein's theory of Relativity. However, these changes are extremely small and are thus ignored in the program.

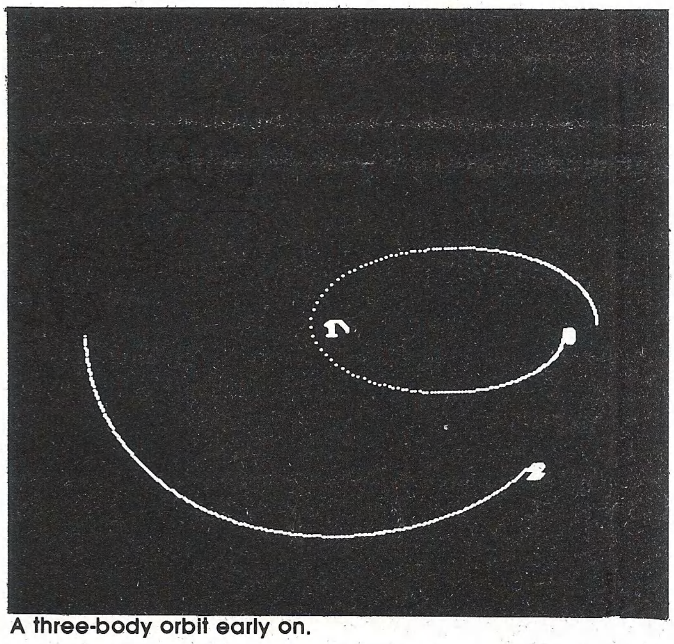
TECHNICAL DETAILS

The program is a fairly straightforward simulation in BASIC. MODE 4 graphics are used to give high resolution plots of the orbits in the standard black on white.

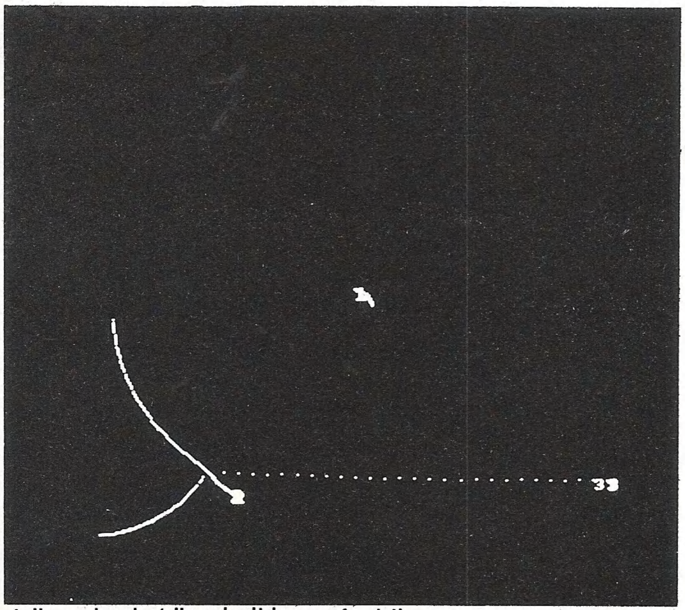
As we only have two colours available in MODE 4, the current positions still have to be identified by plotting numbers (ie 1 for body one, 2 for body two and 3 for body three). The easiest way of achieving this,



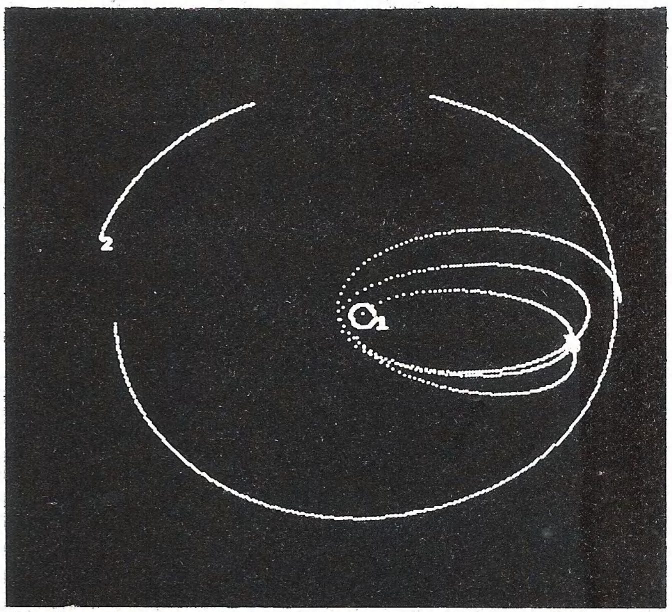
A simple two-body orbit plot.



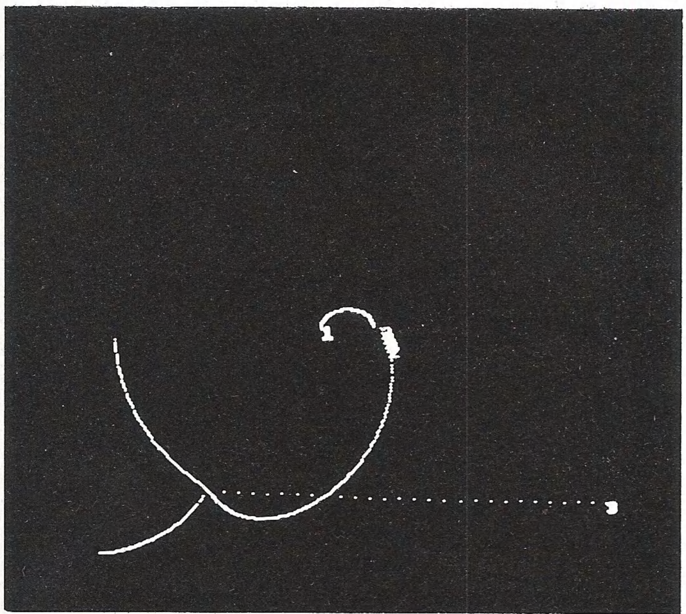
A three-body orbit early on.



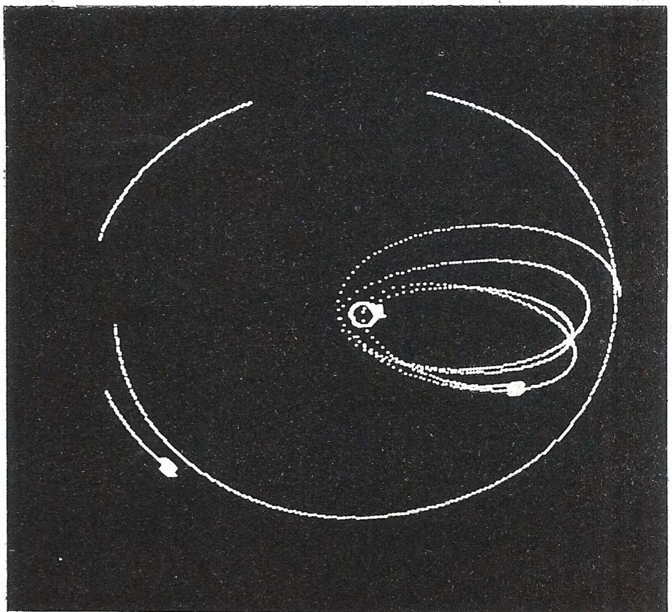
A three-body 'slingshot' type of orbit.



The three-body orbit at an intermediate stage.



As above but after a longer period showing the movement of Body 2.



The three-body orbit showing the displacement of Body 3.

▶

with the full positional accuracy of MODE 4, is to join the text and graphics screens with a VDU 5 command. This allows us to PRINT a character ***anywhere*** on the screen by moving the graphics cursor to the required position (MOVE X,Y) and then simply printing the character (PRINT '1';).

The only trouble with this joining of text and graphics is that normal printing of questions, prompts, etc is slow. To avoid this, the text and graphics screens are not joined until just before the program is ready to plot the orbits.

It is also worth examining the way the PLOT routines use the GCOL command to PLOT and REPLOT the position markers. If you want only the current positions displayed then all you have to do is to unplot the character at the old location by printing a blank and plot the character at the new position by a MOVE and PRINT. The only trouble is that the obvious method doesn't work! When you join the text and graphics screens together you agree to play by the rules of the graphics commands rather than the text commands. Printing a blank will not overwrite an existing foreground character. The reason for this is that, unless you specify otherwise, only the foreground colour is printed and the blank contains no foreground (white) points!

If you would like a demonstration of this fact, simply join the graphics and text screens together and print a number of different characters at the same place — you will see that once a foreground colour is plotted it is never removed and the different characters become jumbled.

It looks as though this 'playing by graphics rules' is going to be a problem but we ***can*** use it to our advantage. If before plotting the position marker we use a GCOL 3,1 instruction, this will form the EXCLUSIVE OR of the plotting colour and any colour that is already present.

This may sound difficult but as MODE 4 has only two colours this reduces to the following rule: if you plot a foreground point (white) and the point is already a foreground colour (white) then the result is the background colour (black) and if you plot a foreground point and the point is a background colour then the result is a foreground colour.

In other words, the first time you plot a character at a blank location it appears and the second time it disappears; this is, of course, exactly what we want for a plot-unplot routine!

We can make use of one further tricky GCOL command to make sure the position markers do not blank out the continuous orbit plots. If you request a continuous display of positions, a point is plotted before the position marker is moved. This point is plotted after a GCOL 0,1 command which forces the point to appear

PROGRAM STRUCTURE

Statement	Function	Action
Lines 10-160	Start up	Selects MODE 4 and defines arrays used in program. Prints title and asks whether full instructions are required.
Lines 170-370	Instructions	Prints explanation of program.
Lines 380-580	Two Body Information	Asks whether a two or three body simulation is required. Prompts for input about the two bodies.
Lines 590-750	Initial Position/ Velocity	Calculates the position and velocity of each of the two bodies so that they orbit in the centre of the screen, adjusting the value of the gravitational constant so that they move at a sensible rate. The positions and velocities of the two bodies are displayed.
Lines 760-860	Third body	The position and velocity of the third body, if specified, is requested.
Lines 870-910	Alter Velocities	The velocities of all three bodies are adjusted to ensure that the centre of mass is stationary.
Lines 920-1020	Display Mode	Offers choice of two types of display.
Lines 1030-1200	Scaling	The screen is cleared, the scaling variables are set, the screen size is determined and the text and graphics areas of the screen are joined.
Lines 1210-1530	Move Routines	Bodies are placed on the screen by scaling and rounding the position x,y to produce a character cell P1,Q1 on the screen. The velocities of the bodies is re-calculated taking account of the gravitational attraction due to the other bodies over a small time interval. The bodies are then moved a distance proportional to their velocity.
Lines 1540-1550	End	'Simulation finished' message.
Lines 1560-1750	Plot Routines	Updates screen by either removing old positions and adding new ones or simply by adding new ones.

no matter what its original colour was. When the position marker is plotted, a GCOL 3,1 call ensures that it cannot wipe out its own trail by making any trail points that clash with position marker points change to the background colour.

When the position marker is plotted for the second time the trail point is restored to its original colour!

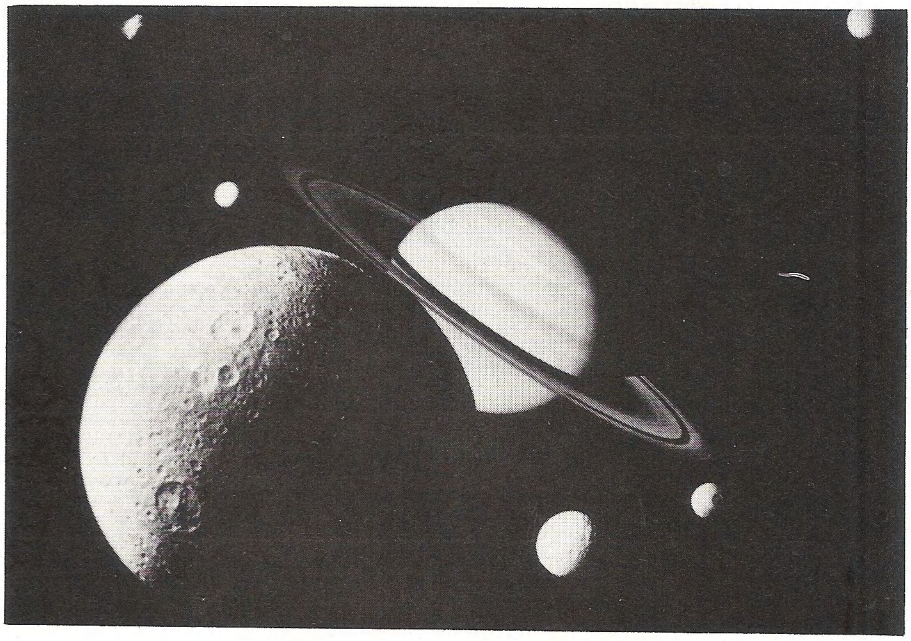
HOW TO USE THE PROGRAM

The program will ask first if full instructions are required. If you answer 'YES' an explanation of the program is displayed. Next, you will be asked if you wish to study the motion of two or three cosmic bodies.

For simplicity, the first two bodies are always placed in motion around each other, and it is only necessary to specify the ratio of the masses of the two bodies and the distance between them. The computer will ask you to input these values. The program then calculates the position and velocity of each of the two bodies so that they orbit in the centre of the screen. The program adjusts the value of the gravitational constant, G, so that the bodies move at a sensible rate on the screen. The position and velocity of each of these two bodies is printed to assist you in choosing sensible values for the position and velocity of the third body if these bodies are specified.

If a third body is being considered, then its position (x,y) and the x,y components of its velocity are requested together with its mass. Care must be taken when choosing these values or the third body will shoot off the screen in a desperate attempt to reach infinity!

When the computer calculates the position and velocity of the first two bodies, it does so in such a way that their centre of mass is stationary and they are positioned at the centre of the screen; this ensures that the bodies stay on the screen. The addition of a third body not only moves the centre of mass but also sets it moving so that the picture could drift off the screen. However, to prevent this happening the program adjusts all the velocities to make the centre of mass stationary.



During the simulation, the program calculates the positions of the cosmic bodies at regular intervals displaying them in real time on the screen. Two different display modes are available:

1. In this mode, the paths traced by each of the bodies (1, 2 and perhaps 3) are displayed. This allows the route traversed by the bodies to be studied, but the screen may gradually become so filled that it may become confusing.
2. In this alternative mode, only the current positions of the bodies are displayed and, as new positions are calculated, the old ones are erased.

You will be asked to make the choice between these two modes.

Sample Data

The following data may be useful to get a feel for the program.

(a) 2 bodies
    mass of second body = 0.3
    distance apart = 10

This produces two stable non-intersecting ellipses.

(b) 3 bodies
    mass of second body = 0.05
    distance apart = 10
    third body:
    position = 10,0
    velocity = 0,5
    mass = 0.01

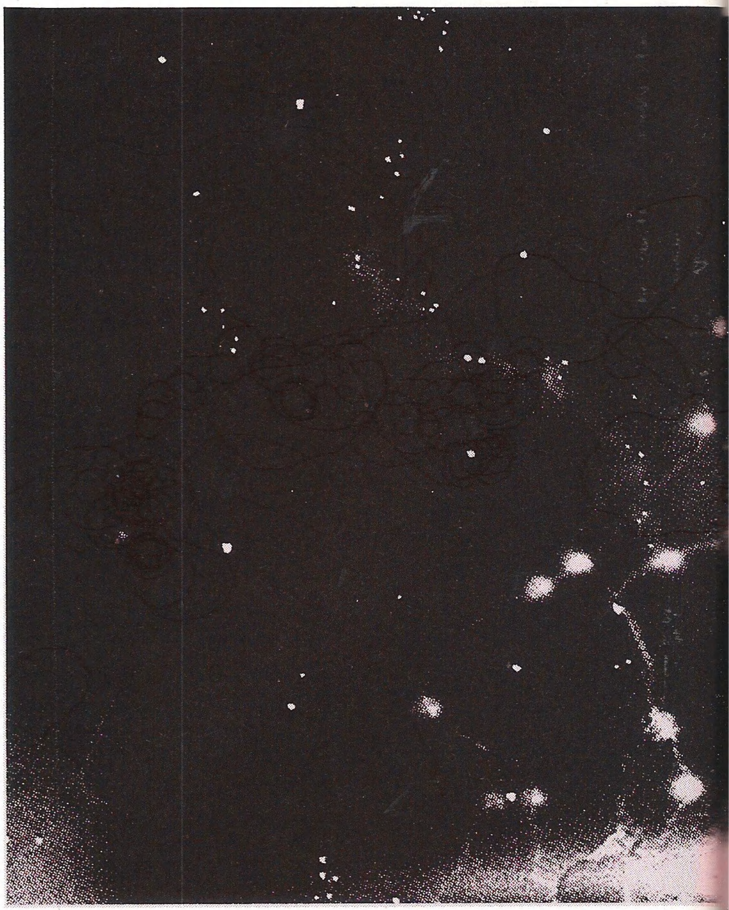
Body 2 follows a large slow elliptical orbit while body 3 follows a tight fast elliptical orbit. Body 1 moves only slightly because of its large mass. Note that body 3 starts at only half of the speed of body 2 but body 3 completes a revolution in less time. This illustrates an interesting principle of spacecraft docking. To catch up with a spacecraft in front of you, you must slow yourself down! Enthusiasts who try speeding up their ship merely fly off into a larger orbit and take longer to complete a revolution. Such hotheads are thus even further behind after one revolution! The correct method of docking is to slow down and dock after one revolution.

(c) 3 bodies
    mass of second body = 0.1
    distance apart = 10
    third body:
    position = -10,-10
    velocity = 3,0
    mass = 0.01

This illustrates a 'sling-shot' in which body 3 gains great speed by means of an encounter with body 2. Body 3 disappears rapidly from the screen, leaving bodies 1 and 2 orbiting each other. These orbits gradually drift across the screen in the opposite direction to body 3. ▶

```
 10  MODE  4
 20  CLS
 30  DIM X(3),Y(3),A(3),B(3),P(3),Q(3),M(3),
     R$(10),C$(1)
 40  REM**X,Y ARE POSITIONS, A,B  ARE
     VELOCITIES, P,Q ARE PLOTTED POSITIONS
 50  PRINT
 60  PRINTTAB(10);"ORBITAL MOTION"
 70  PRINT TAB(10);"=============="
 80  PRINT
 90  PRINT "Would you like FULL instructions"
100  PRINT
110  PRINT "Type YES or NO and press RETURN";
120  INPUT R$
130  IF R$="YES" THEN 170
140  IF R$="NO"THEN 380
150  PRINT "Reply '"R$"' not understood.";
160  GOTO 110
170  CLS:PRINT TAB(0,3);"This program
     analyses the"
180  PRINT"the motion of cosmic bodies"
190  PRINT"relative to each other. This may be"
200  PRINT"used to show the orbits of planets"
210  PRINT"around a single sun, or the
     complex"
220  PRINT"orbits of a planet around a
     double star"
230  PRINT"system. The motion of these bodies"
240  PRINT"is displayed on the screen"
250  PRINT"You must specify whether to
     analyse two"
260  PRINT"or three bodies. The first two are"
270  PRINT"automatically placed in
     mutual orbit"
280  PRINT"around each other, but the
     third body"
290  PRINT"may be placed in any position and"
300  PRINT"with any velocity.
     Considerable care"
310  PRINT"is needed to avoid the third body"
320  PRINT"shooting off to infinity!
     The program"
330  PRINT"automatically adjusts the initial"
340  PRINT"velocities so that the
     centre of mass"
350  PRINT"does not move off the screen"
360  PRINT TAB(0,30);"PRESS ANY KEY TO
     CONTINUE";:A$=GET$
370  CLS
380  PRINT
390  PRINT "Would you like a two body or
     a three"
400  PRINT "body analysis"
410  PRINT
420  PRINT "Type 2 or 3 and press RETURN";
430  INPUT N
440  PRINT
450  IF (N-2)*(N-3)=0 THEN 480
460  PRINT "The number of bodies
     analysed must be 2 or 3"
470  GOTO 420
480  PRINT:PRINT "Consider two bodies
     orbiting each other."
490  PRINT "Let the Mass of the first be one,"
500  PRINT "type the Mass of the second";
510  INPUT M(2)
520  IF M(2)<=0 THEN 490
530  LET M(1)=1
540  PRINT:PRINT "Type in the distance
     between the bodies";
550  INPUT D
560  IF D>0 THEN 600
570  PRINT "Distance must be greater than zero"
580  GOTO 540
590  REM**CALCULATE INITIAL POSITIONS
     AND VELOCITIES OF THE TWO PARTICLES
600  LET Y(1)=0
610  LET Y(2)=0
620  LET X(1)=D*M(2)/(M(1)+M(2))
630  LET X(2)=-D*M(1)/(M(1)+M(2))
640  LET A(1)=0
650  LET A(2)=0
```

```
660  LET G=D*D*D/(M(1)+M(2))
670  LET B(1)=X(1)*SQR(G*(M(1)+M(2))/D^3)
680  LET B(2)=X(2)*SQR(G*(M(1)+M(2))/D^3)
690  PRINT
700  FOR I=1 TO 2
710  PRINT "Body";I
720  PRINT "Position (";X(I);",0)"
730  PRINT "Velocity (0,";B(I);")"
740  PRINT
750  NEXT I
760  IF N=2 THEN 920
770  PRINT "Type position (X,Y) of the
     third body"
780  INPUT X(3),Y(3)
790  PRINT "Type velocity (X,Y) of the
     third body"
800  INPUT A(3),B(3)
810  PRINT "Type Mass of third body.
     (Mass of body 1 is 1)"
820  INPUT M(3)
830  IF M(3)>0 THEN 870
840  PRINT "Mass must be greater than zero"
850  GOTO 810
860  REM**CHANGE VELOCITIES SO THAT
     CENTRE OF MASS IS STATIONARY
870  LET M=M(1)+M(2)+M(3)
880  FOR I=1 TO N
890  LET A(I)=A(I)-A(3)*M(3)*M(I)/M
900  LET B(I)=B(I)-B(3)*M(3)*M(I)/M
910  NEXT I
920  PRINT"PRESS ANY KEY TO CONTINUE":
     A$=GET$
930  CLS:PRINT:PRINT "During the
     simulation,would you like displayed"
940  PRINT
950  PRINT"    1) The PATH traced by the
     bodies"
960  PRINT
970  PRINT "or 2) The CURRENT position
     of the bodies"
980  PRINT
```

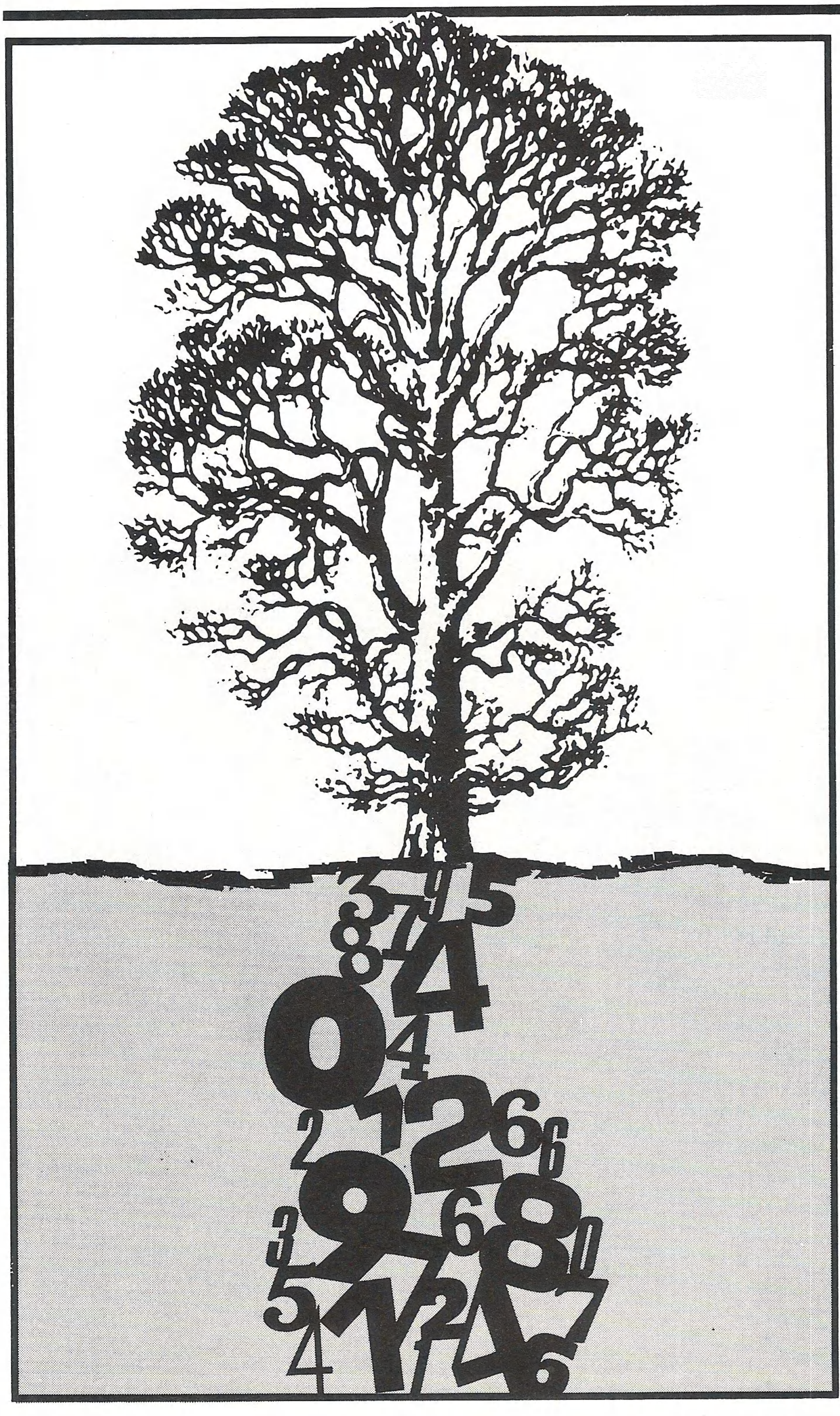


```
 990  PRINT "Type 1 or 2 and press RETURN";
1000  INPUT C
1010  IF(C-1)*(C-2)<>0 THEN 920
1020  REM**CLEAR SCREEN
1030  CLS
1040  REM**SET INITIAL VARIABLES
1050  LET H1=0
1060  LET T=.01
1070  FOR I=1 TO N
1080  LET P(I)=0
1090  LET Q(I)=0
1100  IF ABS(X(I))<H1 THEN 1120
1110  LET H1=ABS(X(I))
1120  IF ABS(Y(I))<H1 THEN 1140
1130  H1=ABS(Y(I))
1140  NEXT I
1150  REM**SPECIFY HEIGHT AND WIDTH OF SCREEN
1160  H=1024
1170  W=1280
1180  VDU 5
1190  LET H0=.5*(H-1)/H1
1200  LET W0=.5*(W-1)/H1
1210  FOR L=1 TO 5000
1220  REM**POSITION BODIES
1230  FOR I=1TO N
1240  REM**CONVERT X,Y POSITION TO P,Q
      CHARACTER CELL
1250  LET P1=INT((X(I)+H1)*W0+.5)
1260  LET Q1=INT((Y(I)+H1)*H0+.5)
1270  IF ABS(P1-P(I))+ABS(Q1-Q(I))=0 THEN 1300
1280  REM**CALLSUBROUTINE TO UNPLOT OLD
      SQUARE AND PLOT NEWSQUARE
1290  GOSUB 1580
1300  NEXT I
1310  REM**RE-CALCULATE VELOCITIES OF BODIES
1320  FOR I=1 TO N-1
1330  FOR J=I+1TO N
1340  LET X1=X(I)-X(J)
1350  LET Y1=Y(I)-Y(J)
1360  LET R=SQR(X1*X1+Y1*Y1)
1370  LET R3=G*T/(R*R*R)
1380  REM**CALCULATE CHANGE IN VELOCITY
      IN X DIRECTION
1390  LET F=X1*R3
1400  LET A(J)=A(J)+F*M(I)
1410  LET A(I)=A(I)-F*M(J)
1420  REM**CALCULATE CHANGE IN VELOCITY
      IN Y DIRECTION
1430  LET F=Y1*R3
1440  LET B(J)=B(J)+F*M(I)
1450  LET B(I)=B(I)-F*M(J)
1460  NEXT J
1470  NEXT I
1480  REM**MOVE BODIES A DISTANCE
      PROPORTIONAL TO VELOCITY
1490  FOR I=1 TO N
1500  LET X(I)=X(I)+A(I)*T
1510  LET Y(I)=Y(I)+B(I)*T
1520  NEXT I
1530  NEXT L
1540  PRINT "Simulation finished"
1550  STOP
1560  REM**SUBROUTINE TO UNPLOT AND REPLOT
1570  REM**UNPLOT
1580  MOVE P(I),Q(I)
1590  IFP1*(P1-W+1)>0 THEN RETURN
1600  REM**REPLOT IF POINT ON SCREEN
1610  IFP1*(P1-W+1)>0 THEN RETURN
1620  IF Q1*(Q1-H+1)>0 THEN RETURN
1630  LET P(I) =P1
1640  LET Q(I)=Q1
1650  LET C$="3"
1660  IF I=3 THEN 1700
1670  LET C$="2"
1680  IF I=2 THEN 1700
1690  LET C$="1"
1700  PRINT C$;
1710  MOVE P(I),Q(I)
1720  IF C=1 THEN GCOL 0,1:PLOT 65,0,0
1730  GCOL 3,1
1740  PRINT C$;
1750  RETURN
```

Converted from a program by D V Mercy

ROOT SOLVER

Even experienced mathematicians can baulk at the idea of solving high-order polynomials, but type in this program and you'll have a head start.



This program calculates and lists the roots of polynomials (provided the coefficients are real). The polynomial can be of any order and both real and complex roots are derived.

Back To Roots

An iteration procedure, based on the method devised by Lin and Bairstow, is used to extract quadratic factors from the polynomial being considered. This procedure has good convergence characteristics and is nearly always successful (even when the initial trial factor is wildly wrong). In the case of an odd order polynomial, there is at least one real root and this is determined last of all.

Generally speaking, the procedure is to start with a trial quadratic factor and to divide it into the polynomial under investigation. If, by good luck or insight, the quadratic is an exact factor of the polynomial then no remainder is left after the division has been carried out. However, as is more likely, a remainder will be left, in which case the Lin-Bairstow iteration is used repeatedly to optimise the coefficients of the quadratic factor. After each trial division the magnitudes of the remainder are tested and if sufficiently small, that cycle of iteration is terminated. Well-known methods are available to obtain two roots from the quadratic factor so obtained.

If the initial polynomial was of order n, then after the quadratic factor has been extracted, the polynomial remaining is or order (n-2). The process is repeated to find another factor, until the final polynomial left is simply first order or a quadratic which is easily solved.

The program gives the user the opportunity to enter trial values of his or her own before the iteration is started. However, this may reduce the execution time of the program. Failing this, each new iteration starts with a trial quadratic factor of $x^2 + 0x + 0$.

True Or False

Although in principle, polynomials of any order can be solved, rounding errors will become important with high order polynomials. Also, because the procedure computes the roots sequentially and the accuracy of latter roots will depend on the earlier ones, errors can accumulate. This is especially true for polynomials with repeated roots.

So take great care before you treat the results of this program as true. Put the roots back into the equation and see how close to zero it actually is!

If you wish to delve deeper into the mathematics used in the program, the following references will help you:

W Hamming, **Numerical Methods**, published by McGraw Hill, 1962.
J L Mueller, **Arithmetic Techniques**, Electronic Design 12.9.68.
Beech et al, **Computer Programs that Work**, Sigma Press, 1978.

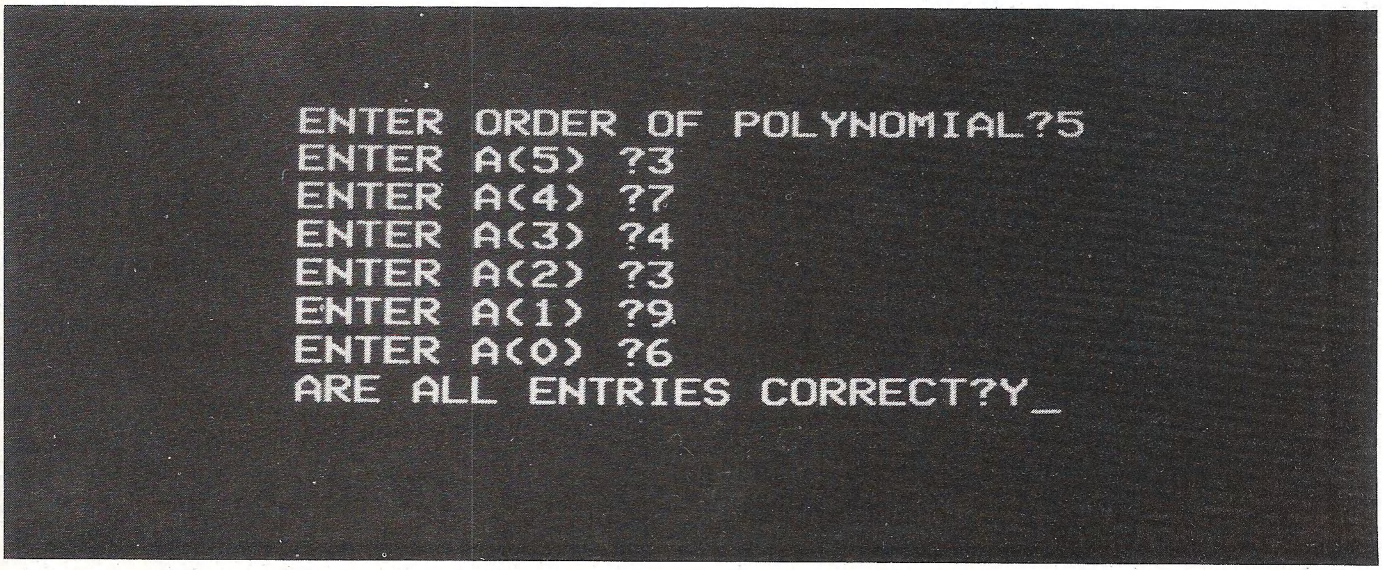
HOW TO USE THE PROGRAM

The program asks first for the order — ie the highest power of x — of the polynomial. Then each coefficient is entered starting with the higest power and the computer allows you to make any necessary corrections before it commences the task.

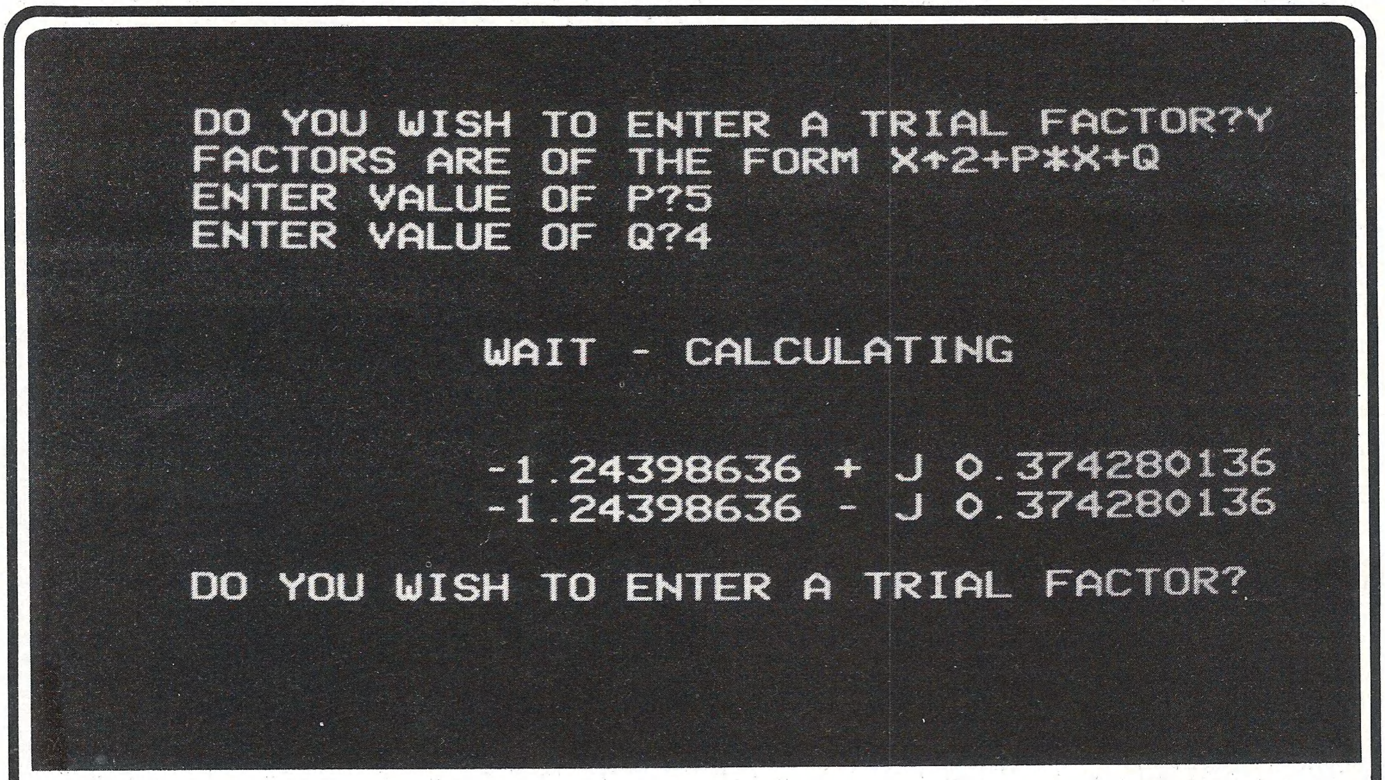
Also, before the program begins the calculation, it asks if you want to enter a trial solution. If the answer is 'NO' the roots are printed as they are found. This can take some time so be patient! If you do opt for entering a trial solution you are asked for the coefficients of a quadratic equation for each quadratic factor the program uses.

PROGRAM STRUCTURE

Statement	Function	Action
Lines 10-220	Input	User is requested to input order of the polynomial. The computer then prompts for each coefficient and allows user to correct input. If any entry is invalid, the computer requests correction.
Lines 230-360	Check	Checks for simple solutions.
Lines 370-480	Preparation	Sets up the inital conditions for iteration and asks user for trial values if required.
Lines 490-560	Division	Carries out synthetic division procedure.
Lines 570-680	Test Remainders	Inspects remainders. If these are small enough the program goes to the print routine, otherwise updates the coefficients of the quadratic factor and carries out synthetic division repeatedly.
Lines 690-800	Print	Prints the quadratic solution.
Lines 810-860	Move Solutions	Tests whether any more solutions are required.
Lines 870-890	Failure	In the event of program halting, prints error message.



```
ENTER ORDER OF POLYNOMIAL?5
ENTER A(5) ?3
ENTER A(4) ?7
ENTER A(3) ?4
ENTER A(2) ?3
ENTER A(1) ?9
ENTER A(0) ?6
ARE ALL ENTRIES CORRECT?Y_
```



```
DO YOU WISH TO ENTER A TRIAL FACTOR?Y
FACTORS ARE OF THE FORM X↑2+P*X+Q
ENTER VALUE OF P?5
ENTER VALUE OF Q?4


                WAIT - CALCULATING


                -1.24398636 + J 0.374280136
                -1.24398636 - J 0.374280136

DO YOU WISH TO ENTER A TRIAL FACTOR?
```

```
 10  REM**ROOTS OF POLYNOMIALS
 20  REM**ROOTS OF A(N).X^N +
     A(N-1).X^(N-1)+...+A(0)
 30  CLS
 40  REM**INPUT DATA
 50  PRINT:INPUT "ENTER ORDER OF POLYNOMIAL",N
 60  IF ABS(INT(N))=N THEN 80
 70  PRINT "INVALID ENTRY":GOTO 50
 80  DIM A(N+2),B(N+2),C(N+2)
 90  FOR M=N TO 0 STEP -1
100  PRINT "ENTER A(";M;") ";
110  INPUT A(M)
120  NEXT M
130  INPUT "ARE ALL ENTRIES CORRECT",K$
140  IF LEFT$(K$,1)="Y" THEN 230
150  IF LEFT$(K$,1)<>"N" THEN 130
160  INPUT "ENTER ORDER OF INCORRECT COEFF.",L
170  IF L>N THEN 190
180  IF ABS(INT(L))=L THEN 200
190  PRINT "INVALID ENTRY":GOTO 160
200  PRINT
210  PRINT "ENTER NEW VALUE OF A(";L;")";
220  INPUT A(L): GOTO 130
230  IF N=0 THEN PRINT "TRIVIAL
     PROBLEM": GOTO 890
240  IF A(N)=0 THEN N=(N-1):GOTO 230
250  FOR M=(N-1) TO 0 STEP -1
260  A(M)=A(M)/A(N):NEXT M:A(N)=1:H=0
270  CLS
280  FOR W=1 TO 1000:NEXT W
290  CLS
300  REM**TEST FOR THE ROOTS AT THE ORIGIN
310  IF A(0)<>0 THEN 340
320  PRINT "0":N=N-1:IF N=0 THEN 890
330  FOR M=0 TO N:A(M)=A(M+1):NEXT M:GOTO 300
340  REM**TEST FOR 1ST OR 2ND ORDER
     POLYNOMIAL
350  IF N=1 THEN PRINT TAB(10);-A(0):
     GOTO 890
360  IF N=2 THEN P=A(1):Q=A(0):GOTO 720
370  REM**SET UP INITIAL CONDITIONS FOR
     ITERATION
380  A(N+2)=0:A(N+1)=0:B(N+2)=0:B(N+1)=0
390  C(N+2)=0:C(N+1)=0
400  P=0:Q=0:R=0:S=0:K=1E-7
410  IF H=1 THEN 490
420  PRINT:PRINT "DO YOU WISH TO ENTER
     A TRIAL ";
430  INPUT "FACTOR",H$
440  IF LEFT$(H$,1)<>"Y" THEN H=1:GOTO 490
450  PRINT "FACTORS ARE OF THE FORM X^2+P*X+Q"
460  INPUT "ENTER VALUE OF P",P
470  INPUT "ENTER VALUE OF Q",Q
480  PRINT
490  PRINT:PRINT TAB(10);"WAIT - CALCULATING"
500  REM**SYNTHETIC DIVISION PROCEDURE
510  P=P+R:Q=Q+S
520  FOR M=N TO 1 STEP -1
530  B(M)=A(M)-Q*B(M+2)-P*B(M+1)
540  C(M)=B(M)-Q*C(M+2)-P*C(M+1)
550  NEXT M
560  B(0)=A(0)-Q*B(2)
570  REM**TEST REMAINDERS
580  IF ABS(B(0)/A(0))>K THEN 610
590  IF ABS(B(1)/A(1))>K THEN 610
600  GOTO 690
610  REM**UPDATE OF P AND Q
620  D=C(1)-B(1)+P*C(2)
630  E=C(2)+P*C(3)
640  F=D*C(3)-E*C(2)
650  IF F =0 THEN 870
660  R=(B(0)*C(3)-E*B(1))/F
670  S=(D*B(1)-B(0)*C(2))/F
680  GOTO 510
690  PRINT
700  REM**SOLUTION OF QUADRATIC FACTOR
710  PRINT
720  T=P*P-4*Q
730  IF T<=0 THEN 770
740  IF P<0 THEN Z=-P/2+SQR(T)/2:GOTO 760
750  Z=-P/2-SQR(T)/2
760  PRINT TAB(10);Z:PRINT TAB(10);Q/Z:
     GOTO 810
770  IF T<0 THEN 790
780  PRINT TAB(10);-P/2:PRINT TAB(10);-P/2:
     GOTO 810
790  PRINT TAB(10);-P/2;" + J ";SQR(-T)/2
800  PRINT TAB(10);-P/2;" - J ";SQR(-T)/2
810  REM**PRELIMINARIES BEFORE LOOPING BACK
820  IF N<3 THEN 890
830  N=N-2
840  FOR M=N TO 0 STEP -1
850  A(M)=B(M+2):NEXT M
860  GOTO 350
870  PRINT "DETERMINANT = 0"
880  PRINT "PROGRAM HALTED"
890  END
```

BIBLIOGRAPHY

Within weeks of the BBC Micro being launched, a number of books had appeared to support the product, both from the BBC and various independant publishers. While this list is complete at the time of going to press, it does not take much foresight to realise that during the next few months even more titles will be published.

The Computer Book: This publication, although produced by the BBC, has no direct connection with either the BBC Micro or The Computer Programme. It simply attempts to introduce the 'computer shy' individual to the wide and diverse subject of computing, covering the ground in a relaxed and friendly fashion. The book doesn't introduce any radical new ideas and certainly cannot be regarded as a 'text' on computers but then that isn't really its object.

The features which really make this book stand out are its excellent production and layout and the clever use of photographs, illustrations and cartoons to keep the reader both interested and amused. If the quality of the editorial content was to the same standard....

The Computer Book by Robin Bradbeer, Peter de Bono and Peter Laurie is published by BBC Publications at £6.75 for 254 pages.
ISBN 0 563 16484 0.

30 Hour BASIC: If you want to approach the subject of computer programming in a disciplined and methodical fashion then this book, which has been produced in conjunction with the BBC series, is almost certainly a recommended buy. You don't really need a micro to complete the course, although one would be helpful, and the book is not specifically related to the BBC Micro . . . a special version is also available for the Sinclair ZX81.

If you like your approach to computers to be light-hearted, this book will probably not appeal. Also, its approach means that as well as learning the ins and outs of BASIC programming, you will learn to write clear and logical programs, something that happens all to seldom in many books.

30 Hour BASIC by Clive Prigmore is published by The National Extension College at £5.50 for 256 pages.
ISBN 0 86082 269 9.

BASIC Programming on the BBC Microcomputer: To have produced this introductory book in so short a time is a remarkable achievement by both the authors and the publishers. The sad fact, however, is that this is an introductory book, and as such tends to leave you waiting for more. The volume was put together with the assistance of Acorn, the company who designed and produced the BBC Micro, so it is very specifically related to that product.

The book is practical in nature with lots of small examples to try out and problems to solve. The actual information content is not significantly more than that in the early version of the User Guide except that the facts have been arranged in a more readable form. Although the book has areas of weakness, it does stand up as an introduction but one hopes that the second volume will not be too long in coming.

BASIC Programming on the BBC Microcomputer by Neil and Pat Cryer is published by Prentice Hall International at £5.95 for 205 pages.
ISBN 13 066407 3.

Practical Programs for the BBC Computer and the Acorn ATOM: This somewhat slimmer volume contains four chapters based around a number of simple programs which are reproduced for both the BBC Micro and the ATOM. The presentation and layout is excellent and the structure of each of the examples is clearly explained. However, the real meat is to be found in the fifth chapter which presents SPL, Simple Programming Language, a new compiler for both types of micro. As well as providing a second high-level language, this chapter demonstrates how to go about writing a simple compiler.

Practical Programs for the BBC Computer and the Acorn ATOM by David Johnson-Davies is published by Sigma Technical Press (distributed by John Wiley & Sons Ltd) at £5.95 for 120 pages.
ISBN 0 905104 14 5.

Coming Soon

A large number of book publishers have material in various stages of development for the BBC Micro user and within the next few months we can expect titles on the graphics capabilities of the system, its music and sound generation, at least two more titles on the BASIC it uses plus several more general works. The flood of books is somewhat akin to the one which took place after the launch of the ZX80 and ZX81 but, so far at least, the quality seems to be of a much better standard.

The two titles I haven't covered are the two versions of the User Guide itself. Suffice to say that everything the first version lacked (simply because of the time needed to print it before the machine was ready), the second version seems to make up for. Indeed, the second version may well set the kind of standard demanded of future systems, that is if the manufacturers can be persuaded to give all their secrets away!

TIRED OF TYPING?

Why type in 77,000 bytes of BBC BASIC when you can buy these programs all ready to be LOADed into your BBC Micro.

Breakthrough — You have a bat, five balls and a wall of six rows of bricks. The aim is to break through the barrier of bricks scoring as many points as possible. A classic game and quite compulsive.
Ski-Run — Downhill slalom complete with gates to negotiate. Reach the bottom of the course in the shortest possible time, steering your way around the gates and without going off the edge of the course. You may alter the speed as you go, but even slow is too fast for some!
Alien Attack — Control the position of your space ship so that you can shoot down the invading aliens. They seem to travel slowly but they never stop!
Surround — The classic game for the PET computer now rewritten for the BBC Micro. The object is to move your piece around a moving 'enemy' so that he has nowhere to go! Fast reactions are needed for this one.
Crystal Ball — Want some fun with your friends? Type in their name and get the computer to generate a lighthearted character assessment based on a reduction of the subject's name to a single 'magic' number.
Battle of Britain — Based on two months at the height of the Battle of Britain, you allocate the number of British radar sites to be implemented, the number of fighters to scramble, the number of fighters to manufacture, etc. The computer plays the German side. Plan your campaign carefully or in the world of the BBC Micro, we may all be speaking German!
Fox and Hounds — A game played on a chess board grid where the computer controls the fox and you control the hounds. The aim is to trap the fox. Good graphics and a game with varying degrees of skill.
The White Barrows — Trap the Sorcerer and recover the Magic Chalice, using your magical powers carefully to negotiate and locate his position. Bribe goblins, fight dragons and work out the layout of the labyrinth. A good text-only game which should keep the family spellbound for hours.

Squish — No game of squash was ever like this! Between you and the back wall are a row of pillars which the 'ball' can bounce off! Great fun and quite addictive.
Obstacle Course — Race against the clock to get through a maze which is continually changing its pattern. Grit your teeth when it blocks you but don't despair, you'll get through in the end. But how much time you'll take is anyone's guess.
Dambuster — Watch those bombs bounce as they hit the water . . . will you destroy the dam this time? Careful judgement of the height of your plane is essential, the bomb could even bounce back and hit you!
Patterns — A demonstration not only of the BBC Micro's graphics capabilities but also of how restful moving colour images can be. Several choices of pattern including some fantastic web designs which continuously alter. Who wants to go for trancendental meditation when you have Patterns on your own TV screen.
Conquering Everest — Have you ever dreamt of climbing our world's highest mountain? Well, now is your chance to plan out an entire expedition in your own home. This impressive text-only game allows you to travel from base to base, choosing whether you want porters or Sherpas to carry all your provisions. But don't expect to conquer Everest at one sitting, you have to move the right equipment to the right place at just the right time. Have fun . . . without frostbite!
Wordsquare — The idea behind this game is to give the computer a list of words which it then hides in a 15 by 15 grid of letters. They may be hidden in a number of different ways and as the additional letters are randomly selected, no two puzzles are ever the same.
Leapfrog — A computer implementation of the game often found in pubs. The aim of the game is to get the left-hand rows of 'coins' over to the right-hand side and vice versa. Sounds easy . . .

BBC TAPE 3

Home Finance — Use this program to keep track of your finances. The computer will help you to budget by doing the sums and presenting you with all the up-to-date information you need to keep within your spending limits. The program will handle up to 20 accounts.

Investment Checker — Having saved some money using the program above, this one will help you work out what your profits would amount to should you invest a regular amount over a period of time.

Meter Minder — This package provides you with a series of projections based on your meter readings . . . gas and electricity. Fuel consumptions in cost per week, month or quarter. Plan how to SAVE IT.

Calendar — With this versatile program you need never be without a date! The program will give the number of days between any two dates, what day of the week any date fell on and a display of any month of any year since 1582!

Plot 'n' Planner — Work out the cost of your new carpet or how much grass seed you'll need for the lawn. 'Draw' any irregular shape on a displayed grid pattern and the computer will work out the enclosed area.

Base Changer — Do you have trouble converting to and from binary or hexadecimal? This program takes away the guesswork, converting to and from any number base up to 16.

Morse Trainer — Practise Morse Code to perfection with this program. It will help you improve receiving speeds and aid initial recognition of Morse characters, introducing you to Morse Code gradually through a series of planned lessons.

Root Solver — This program will calculate and list the roots of any polynomial (providing the coefficients are real!). The polynomial can be of any order and both real and complex roots are derived.

Newton's Law — With this program, the computer enables the user to get the feel for Newton's Law of Gravitation as applied to the motion of planets. The orbits of planets are calculated and displayed on the screen. It can even simulate the 'slingshot' effect as used to send the Voyager spacecraft from Jupiter to Saturn!

Descriptive Statistics — As statistics packages go, this is a short and simple one — but it will still turn unwieldy raw data into neat statistical information.

Fill in the coupon below
and send it to:
CT SOFTWARE,
ASP Ltd,
145 Charing Cross Road,
London WC2H 0EE.

The programs included in this issue of Personal Software are presented on three tapes, the first two covering the GAMES programs and the third containing the DOMESTIC, TECHNICAL, EDUCATIONAL and FINANCIAL programs.

CT SOFTWARE,
ASP Ltd,
145 Charing Cross Road,
London WC2H 0EE.

BBC TAPE 1 (Games) ☐
BBC TAPE 2 (Games) ☐
BBC TAPE 3 (Domestic, Technical, Education and Finance) ☐

Tapes are priced at £9.95 each or you can order two tapes for £17.50 or all three for only £25! All orders must include an additional 50p for postage and packing.

I enclose my Cheque/Postal Order/International Money Order for: (delete as necessary)
£. . . . .(made payable to ASP Ltd)
OR debit my Access/Barclaycard
(delete as necessary)

☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Please use BLOCK CAPITALS
Name (Mr/ Mrs/ Miss) . . . .
Address . . . .
. . . .**Postcode** . . . .
Signature . . . . **Date** . . . .
Please allow 21 days for delivery

Computing Today Software

BBC VERSION
NOW AVAILABLE

THE VALLEY

© Computing Today 1982

What are you . . .
Barbarian or Wizard?

Choose your character type carefully . . . Barbarians recover quickly but their magic doesn't come easily. A Wizard? Slow on the draw and slow to mature . . . but live long enough and grow wise enough and your lightning bolts are almost unstoppable . . .

The Valley is a real-time game of adventure and survival. You may choose one of five character types to be your personal 'extension of self' to battle and pit your wits against a number of monsters. Find treasure, fight a Thunder-Lizard in the arid deserts of the Valley, conquer a Kraken in the lakes surrounding the dread Temples of Y'Nagioth or cauterise a Wraith in the Black Tower. In fact live out the fantasies you've only dared dream about. BUT BEWARE . . . more die than live to tell the tale!

You've read the program (Computing Today — April '82) . . . Now buy the tape. PET. TRS-80, BBC and Sharp tapes are available at £9.95 per tape plus 50p postage and packing. 16K minimum . . . Commodore PET (New ROMs), TRS-80 Model 1, Level 2, BBC Model B and Sharp MZ-80K. Fill in the coupon below and return it to CT Software, ASP Ltd., 145 Charing Cross Road, London WC2H 0EE and become one of the first to play . . . . The Valley . . . .

Please send me . . . tape(s) of The Valley ☐ PET ☐ TRS-80 ☐ Sharp ☐ BBC at £9.95 per tape plus 50p postage and packing.
I enclose my Cheque/Postal Order/ International Money Order for: (delete as necessary)
£. . . . . (Made payable to ASP Ltd)
OR Debit my Access/Barclaycard
(delete as necessary)

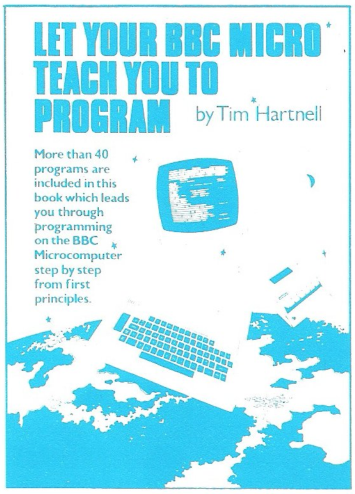
☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Please use BLOCK CAPITALS
Name (Mr/Mrs/Miss) . . . .
Address . . . .
. . . . Postcode . . . .
Signature . . . . Date . . . .
Please allow 21 days for delivery
PS1

Computing Today Software

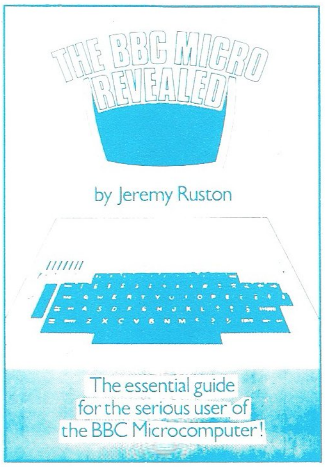
Make the most of your BBC Micro

"Let your BBC Micro teach you to program". This great new book by best-selling author Tim Hartnell is the ideal companion for you if the BBC Micro is your first computer. The book leads you through programming in **BBC BASIC**, *step by simple step, and gives you a host of worthwhile programs to try. As well as a large collection of games (including* **DRAUGHTS** *and* **OTHELLO**), *there are programs for educational, mathematical and statistical applications. You can now master programming quickly, with this great new book. £6.45.*


LET YOUR BBC MICRO TEACH YOU TO PROGRAM
by Tim Hartnell
More than 40 programs are included in this book which leads you through programming on the BBC Microcomputer step by step from first principles.

INTERFACE
The monthly magazine ... and BBC
ZX80, ZX81, Atom
Acorn monthly
Volume 2 Issue 8
INTERFACE
ZX80 ATOM ZX81 BBC
Inside this issue:
Adding joysticks to your ZX81
Floppy disks for the Acorn Atom
Why the BBC put its prices up
The Second ZX Microfair
The ZX and Acorn Monthly
February 1982, Volume 2, Issue 6
ZX80 ZX81 Atom BBC Microcomputer

Each month the Independent Acorn (Atom/BBC) Users' Group publishes the magazine **INTERFACE**, *with major articles and at least six programs for your BBC Microcomputer in each issue. A year's subscription to* **INTERFACE** *is just £9.50 (UK), £12.50 (elsewhere). Just £1.00 will bring you a sample copy.*


THE BBC MICRO REVEALED
by Jeremy Ruston
The essential guide for the serious user of the BBC Microcomputer!

"The BBC Micro Revealed". Jeremy Ruston, author of the popular manual "Pascal for Human Beings", turns his attention to the BBC Microcomputer and reveals all in this great new book. If you're a serious user of the BBC Micro, and you want the full story on what is going on deep in the computer's insides, you need this book. With it, you get the benefits of months of patient probing. £7.50.

INTERFACE, DEPT. PS,
44-46 EARLS COURT
ROAD,
LONDON W8 6EJ

Please send me the following:

☐ The next 12 months of **INTERFACE**. I enclose £9.50 (UK), £12.50 (elsewhere). Please make me a member of the users' club.
☐ A sample copy of **INTERFACE**, and details of the Independent Acorn (Atom/BBC) Users' Group. I enclose £1.00.
☐ A copy of "Let Your BBC Micro Teach You to Program" by Tim Hartnell.
I enclose £6.45.
☐ A copy of "The BBC Micro Revealed" by Jeremy Ruston.
I enclose £7.50.
Total enclosed £. . . . .

NAME . . . .

ADDRESS . . . .

. . . .

BBC MICRO SOFTWARE

from PROGRAM POWER

MODEL A or B

CHESS
High quality chess game with castling, 'en passant', play black or white, rejection of illegal moves and 6 levels of play. Set up problem games. **£7.95**

STARTREK
Full feature version of the classic battle against the Klingons. All the usual features - phasers, photon torpedoes, long and short range scans, 'on-board' computer - together with sound effects and 'real-time' torpedo battles. **£7.95**

MUNCHYMAN
Colourful and highly entertaining version of this popular arcade game. Munch your way to a high score, before the 'munchers' devour you. Reverse roles by munching the stars. The more you score, the harder the game gets. **£5.95**

GOMOKU
Well presented board game, written in Basic and m/c code. The objective is to be first to get five counters in a row. The computer plays a tough game and is very hard to beat, especially when making first move. **£3.95**

DISASSEMBLER
Relocatable program. Lists object code and assembler mnemonics, from any specified address. Stop and restart the listing, page mode option, ASCII symbol and printer output options. **£5.95**

MODEL B ONLY

SPACE MAZE
You have crash landed in the legendary labyrinth of Titan, inhabited by alien monsters known as 'Froogs'. Find your way out to the 'Transmat' probe before being cornered and eaten. 8 skill levels and 3D colour graphics. **£5.95**

WRITTEN ANY PROGRAMS!

WE PAY 20% ROYALTIES FOR ATOM, NASCOM & BBC MICRO PROGRAMS

Special Offer

ORDER BEFORE 30th JUNE AND SAVE £££'s

We are giving Fabulous discounts on **all** our BBC MICRO Programs IF YOU BUY NOW.
CHESS (£7.95) now £4.95.
STARTREK (£7.95) now £4.95. MUNCHYMAN, DISASSEMBLER & SPACE MAZE (£5.95): Now £3.95 GOMOKU (£3.95) now £2.95

Please add 55p order P & P + VAT at 15%

PROGRAM POWER
5 Wensley Road
Leeds LS7 2LX
Tel. [0532] 683186